ENHANCEMENTS OF THE VASTG POST-PROCESSING AND GRAPHICAL DISPLAY PROGRAMS

UNLIMITED DISTRIBUTION

National Defence — Défense nationale
Research and Development Branch — Bureau de recherche et développement

DREA CR/94/417

ENHANCEMENTS OF THE VASTG POST-PROCESSING AND GRAPHICAL DISPLAY PROGRAMS

by
M.F. Palmeter - W.T. Ooi - M.W. Chernuka

MARTEC Limited
1888 Brunswick Street, Suite 400
Halifax, Nova Scotia, Canada
B3J 3J8

CONTRACTOR REPORT

Prepared for

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

Canada

THIS IS AN UNEDITED REPORT ON SCIENTIFIC OR TECHNICAL WORK CONTRACTED BY THE DEFENCE RESEARCH ESTABLISHMENT ATLANTIC OF THE RESEARCH AND DEVELOPMENT BRANCH OF THE DEPARTMENT OF NATIONAL DEFENCE, CANADA.

THE CONTENTS OF THE REPORT ARE THE RESPONSIBILITY OF THE CONTRACTOR, AND DO NOT NECESSARILY REFLECT THE OFFICIAL POLICIES OF THE DEPARTMENT OF NATIONAL DEFENCE.

PLEASE DIRECT ENQUIRIES TO:

THE CHIEF,
DEFENCE RESEARCH ESTABLISHMENT ATLANTIC,
PO BOX 1012,
DARTMOUTH, NOVA SCOTIA,
CANADA.

UNLIMITED DISTRIBUTION

National Defence
Research and Development Branch

Défense nationale
Bureau de recherche et développement

DREA CR/94/417

ENHANCEMENTS OF THE **VASTG** POST-PROCESSING AND GRAPHICAL DISPLAY PROGRAMS

by
M.F. Palmeter - W.T. Ooi - M.W. Chernuka

MARTEC Limited
1888 Brunswick Street, Suite 400
Halifax, Nova Scotia, Canada
B3J 3J8

Scientific Authority Neil Pegg

W7707-0-1174
Contract Number

September 1992

CONTRACTOR REPORT

Prepared for

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

DEFENSIO PER SCIENTIAM

Canada

ABSTRACT

VASTG is an interactive graphics program designed specifically for use with the VAST finite element analysis program and it offers both pre-processing features for verification of the input data and post-processing features for interpretation and presentation of results. Numerous improvements have been made. The model verification modules PLOTV1 and VASHID were upgraded to permit isoparametric curved beams to be plotted as solids and with eccentricities shown. VASHID was also improved in other respects to gain speed and reduce storage requirements. The load data verification capabilities of PLTV12 were expanded. The deformed model plotting capability utilizing VASHID was generalized to permit rotational degrees of freedom for the thick-thin shell elements to be either global or local. In addition, the automatic visible surface identification capability was extended to displacement and mode shape contours. Scaling on finite element plots was permitted to be automatically defined from element plotting specifications. Provisions were made to save graphics generated for terminal display on file for optional output to hardcopy devices such as laser printers. Furthermore, the capability for switching between graphics mode and alphanumerics mode was improved as well.

RÉSUMÉ

VASTG est un logiciel graphique interactif conçu spécialement pour s'utiliser avec le programme d'analyse des éléments finis VAST; il offre à la fois des caractéristiques de pré-traitement servant à la vérification des données d'entrée et des caractéristiques de post-traitement utiles à l'interprétation et à la présentation des résultats. De nombreuses améliorations lui ont été apportées. Les modules de vérification de modèle PLOTVI et VASHID ont aussi été perfectionnés de sorte qu'il soit possible de tracer des faisceaux à courbure isoparamétrique sous forme de solides montrant les excentricités. VASHID a également été amélioré à d'autres égards, de façon à accroître la vitesse et à réduire la capacité de stockage nécessaire. Les fonctions de vérification des données de charge de PLTV12 ont été étendues. Le traçage de modèles déformés au moyen de VASHID se trouve maintenant généralisé, ce qui permet d'appliquer des degrés de liberté de rotation globale ou locale à des éléments de noyau épais-minces. De plus, l'identification automatique des surfaces visibles peut dorénavant s'appliquer aux contours des formes de déplacement et de mode. La mise à l'échelle des tracés à éléments finis pourra se définir automatiquement à partir des spécifications de traçage des éléments. Par ailleurs, il sera possible de sauvegarder les graphiques générés en vue d'un affichage terminal dans un fichier pouvant être dirigé vers un appareil d'impression, comme une imprimante laser. La capacité de commutation entre le mode graphique et le mode alphanumérique a également été améliorée.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Background Comments

VASTG is a graphics package for use in vibration and strength analysis of structures using the finite element method. It offers both pre-processing features, for verification of input data, and post-processing features, for interpretation and presentation of results. VASTG is designed specifically for use with the general purpose finite element computer program VAST [1,2] and so uses directly either input data files prepared for VAST or the output files created by VAST.

VASTG consists of a main program which determines which data files exist and hence what plotting options the user has. The user chooses a particular plotting option from among those presented by the main program, and then enters into a subprogram to do the actual plotting. Once the plotting is complete, the main program then presents the user with a set of options which govern what the program does next.

If the user has finite element model data available which is consistent with the VAST file naming conventions and formats [1,2], options to use the pre-processing graphics programs will exist. Providing the required data exists, the user may generate plots of either structural or fluid finite element models, with control over which nodes and/or elements are displayed as well as control over many other parameters which aid in interpretation of the plot. If data exists for the boundary conditions or lumped masses, these may be indicated on the plot. An option also exists to plot the loads imposed on the finite element model, if the data exists. These may be either concentrated nodal forces or moments or pressure loads applied to individual elements, for either static or dynamic analysis.

The post-processing graphics options presented to the user depend on the type of analysis performed by VAST, and hence the data files available. The post-processing graphics options are quite extensive and include the plotting of deflected shapes from static or dynamic analysis or mode shapes from a vibration or buckling analysis, either alone or superimposed on the undeflected shape. Deflection or mode shape contour lines may also be plotted on the undeflected shape. Displacement time histories from dynamic analyses may be plotted. Frequency response curves from a frequency response analysis may be plotted.

If stresses or strains are to be plotted, the VAST output must first be processed to generate averaged nodal values. Control of the stress or strain post-processing is all handled by VASTG so that the user is guided through the interactive program. The user may plot stress or strain contour lines or principal vectors from static or dynamic analyses, or time history plots from a dynamic analysis. Alternately, the user may desire to process the stress data in a routine VAST analysis batch run. The VAST manual describes this feature in the section on post-processing results. Fatigue failure assessments of stresses from a dynamic analysis may be plotted as well as strain energy density contours from static or dynamic analyses.

VASTG is designed to run interactively, so that any required input is prompted for by the program. The VASTG modules also have the capability to: (i) operate using a session file containing responses to prompts, and (ii) operate using single word command directives. Generally, all input is in free format.

1.2 Current Developments

Because VASTG interfaces directly with VAST input and output files, it must be continuously updated as VAST itself develops. The work discussed in this report is that conducted in parallel with the first phase of the development of a nonlinear, static, large displacement and rotation (geometric nonlinearity) capability within VAST. All work described within this document has been implemented within version 6.0 of VASTG which is compatible version 6.0 of both VAST and VAST-NL.

VASTG serves as a valuable tool in the verification of input data prior to analysis with VAST. PLOTV1 is the model verification module. It plots either structural or fluid finite element models from VAST input data files. Being a data checking program, PLOTV1 offers the user much control over what is shown on the plot. Specified nodes may be plotted and optionally numbered. Similarly, specific elements can be plotted. In the case of structural finite element models, boundary conditions and lumped masses may be displayed.

PLOTV1 was upgraded to permit isoparametric curved beams to be plotted as solids with eccentricities. This enhancement permits the user to verify the position of beams in stiffened plate or shell models. Details are given in Chapter 2.

VASHID is one hidden line program under development within VASTG. It also required improvements with respect to plotting of beam eccentricities. The work performed to implement this capability is described in Chapter 3. Verification of the line element plotting capabilities is discussed in Chapter 4.

While the VASHID algorithms are competitive with alternate hidden line algorithms available within the VASTG framework, careful study of algorithms employed in VASHID revealed some potential improvements were desirable not only with respect to speed but also with respect to array utilization. Various arrays, including the XT array to store middle surface nodes in thick-thin shell elements, were redundant and could be removed from the program with no loss of generality. These and other program enhancements designed to reduce storage requirements and improve computational speed are detailed in Chapter 5.

The VASTG module called PLTV12 plots element loads and concentrated loads for load data verification purposes. Element loads can be either pressure loads, edge loads, point loads (not at nodes) or prestrain axial loads. Concentrated loads can be either nodal forces or moments. In both types of loads, static or dynamic time histories are permitted. Several modifications to PLTV12 were required to upgrade its data verification capabilities for full compatibility with VAST input. Details are supplied in Chapter 6.

VASTG also offers post-processing features which facilitates interpretation and presentation of results. Some enhancements of these features have also been completed in the current contract. The capability to contour displacements or mode shapes on automatically identified visible surfaces has been provided as discussed in Chapter 7. VASHID was modified so that deformed model plots for the thick-thin shell element could be obtained when nodal rotations are defined with respect to global directions. Implementational details are given in Chapter 8. The PLTV16 module which displays beam stress was upgraded to provide the user with greater control on the elements to be displayed. More information on how this increase of flexibility was achieved is given in Chapter 9.

Some refinements are equally significant for pre-processing and post-processing. These are discussed last. Most important, in this context, is the provision for scaling parameters being defined automatically from the element plotting specifications as documented in Chapter 10. Significant changes to the PLOTVX library including the provision of a capability to automatically save graphics output to laser printers and improved control over switches between terminal operation in graphics mode and alphanumeric mode are discussed in Chapter 11. In Chapter 12, are described a number of miscellaneous developments not requiring sufficient effort to warrant individual chapters. As noted in Chapter 13, a documentation file to assist with the installation of the VAST system at new sites has been prepared. Also described in the same chapter is an example problem set which could be employed to validate the installation of the VAST system at new sites. Finally, some improvements made to the VASTG manual are summarized in Chapter 14.

CHAPTER 2

DISPLAY ECCENTRICITIES AND CROSS-SECTIONS OF ISOPARAMETRIC BEAMS

2.1 Introduction

The VASTG graphics module, PLOTV1, is a pre-processing package and serves as a model verification program. Both structural and fluid finite element models can be plotted with the user controlling the nodes and elements to be displayed. The models may optionally be substructured. The capabilities within PLOTV1 to plot VAST isoparametric curved beam element (IEC=7) have been upgraded under this contract. The element was previously plotted only as a line through the displacement nodes which somewhat limited the effectiveness of PLOTV1 in graphical checking of geometric data for the isoparametric curved beam. In particular, orientations and eccentricities could not be checked. Consequently, the capability to plot the isoparametric beam elements with eccentricities when beam nodes correspond with shell nodes has been developed and the capability to plot the element as a solid has also been provided.

2.2 Implementation

The capability to plot the isoparametric beam element (IEC=7) as a solid element required:

(a) Development of a new MARLIB library subroutine CROSS7 to generate coordinates for plotting cross-sections.

(b) Modification to the MARLIB library subroutine ELEMA, which generates the standard model geometry scratch file, to include the coordinates calculated by subroutine CROSS7.

(c) Modification to the PLOTV1 module to:

- i) prompt the user to select type of element plotting desired for this element (line or solid),
- ii) changes to the session file generation subroutine, PLISES, to reflect the new prompt,
- iii) changes to the command subroutine, COMAN1, to include a new command, SOLID, and

2.2

iv) a new section of code to read the coordinates at the cross-section to generate a coordinate list identical to the 20-noded solid. The element plotting subroutine, ELPLOT, is called identifying the beam element as the 20-noded solid (IEC=2).

The capability to plot the isoparametric beam eccentricities required:

(a) Development of a new subroutine AUX7 to generate offsets for the input eccentricities and then to write this data onto the output file.

(b) Modifications to PLOTV1 to read the offset (DX, DY, DZ) and calculate location of the section to be plotted.

2.3 Example

The new plotting capabilities for the isoparametric beam (IEC=7) are demonstrated in Figure 2.1 with the element displayed as a solid element and in Figure 2.2 with the element displayed as a line element.

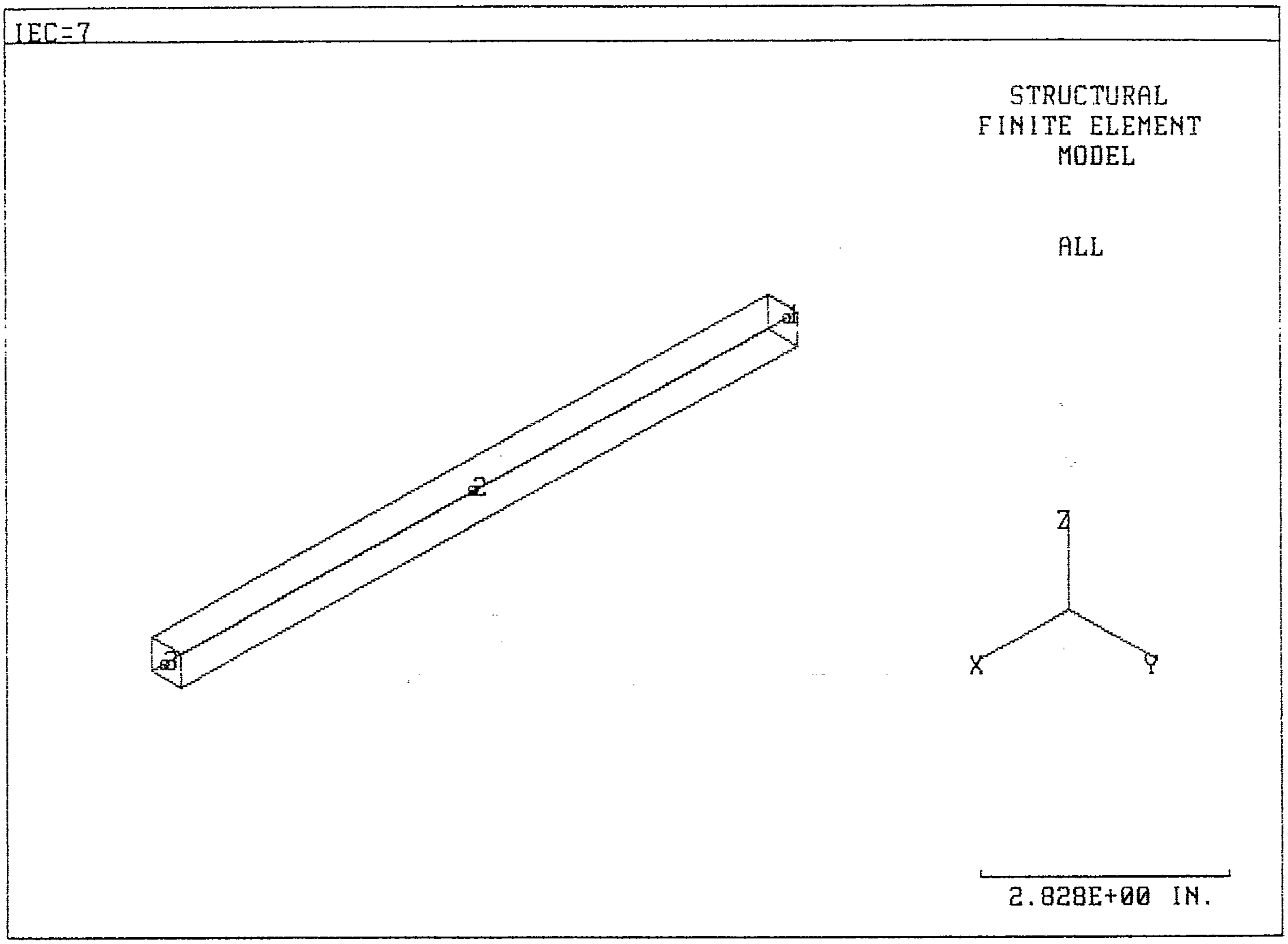


FIGURE 2.1: Curved Beam Element (IEC=7) Plotted as a Solid Element

CHAPTER 3
ACCOUNTING FOR BEAM ECCENTRICITIES IN VASHID

3.1 Introduction

VASHID was originally developed to assess the potential of the hidden line algorithm proposed by J.L. Janssen [3]. This hidden line algorithms was basically very simple and had low array storage requirements. One advantage of the VASHID algorithm over that of the Watkins visible surface scan line algorithm [5] used in MOVIE [6] is that line elements (beams or bars) can be plotted. Consequently, VASHID can be used to verify the positioning of line elements within finite element analysis models when used either independently or in combination with other element types whereas MOVIE cannot. Previously, beam elements were plotted in VASHID without eccentricities shown. This limitation prevents one from verifying the position of beams in stiffened plate or shell models. Some work was carried out to address this deficiency and to properly account for beam eccentricities when plotting beams. This work is described below.

3.2 Program Modifications

Two major modifications were required in VASHID in order to permit the beam eccentricities to be plotted. The first modification involved having offsets of beam eccentricities calculated in the subroutine ELEMA of the MARLIB library for line elements of IEC= 3 and 7 and then written onto NTS file for use in VASHID. The READ statements in VASHID were also correspondingly modified to read the offsets of eccentricities, DX, DY, and DZ in addition to the element connectivities. In the case of curved beam element (IEC=7), the order of nodes in element connectivities being read is also rearranged. In particular, the first node is read first, the centre node second and the third node last. Beam connectivities define the nodes relative to which the supplied offsets are related to. For stiffened shells, the beam connectivities identify the connected shell nodes. In that case, from the element connectivities and the associated beam offsets, new nodes and coordinates can be calculated. Later, in plotting these line elements, the new nodes and coordinates will be used. For models containing only beam elements, the beam connectivities can of course identify beam nodes since the offsets are zero. Therefore, for this case, the generation of new beam nodes is not required and can be avoided by checking to see if the sum of squares of the offsets, DX, DY, and DZ, is less than 0.0001.

The second major modification has involved the enhancement of VASHID plotting capabilities for the finite element analysis models meshed with a combination of different element types. Previously, the VASHID program did not permit the plotting of finite element analysis models meshed with compatible element types in combination such as IEC=1 and 7. This deficiency resulted from the definition of face connectivities for the solid elements (IEC=1,2,6,16,17, and 20) being done in the subroutine FGRID, which was called by the subroutine FCONH, whereas for line elements and membrane/plate elements (IEC=3,4,5,7,8,9,10, and 23) it was done in the subroutine STRAIT. Because the face connectivities for different element types were set up in two separate subroutines, insertion of information about elements into a common data base was not possible. This contract item has addressed this deficiency in the VASHID plotting capabilities. Basically the subroutine STRAIT for definition of the face connectivities for element types of IEC=3,4,5,7,8,9,10, and 23 was called from the subroutine FCONH, instead of from the main VASHID program. This simple modification permitted a finite element analysis model meshed with different but compatible element types to be plotted.

3.3 Example Problems

To illustrate the plotting capabilities of beam elements as stiffeners with the display of eccentricities, two models, have been chosen. Figure 3-1 shows a stiffened shell containing the thick/thin shell element and the curved beam element (IEC=1 and IEC=7). Figure 3-2 shows a stiffened plate comprised of the plate bending element and the general beam element (IEC=3 and IEC=5). In both cases, the beam eccentricities are clearly evident.

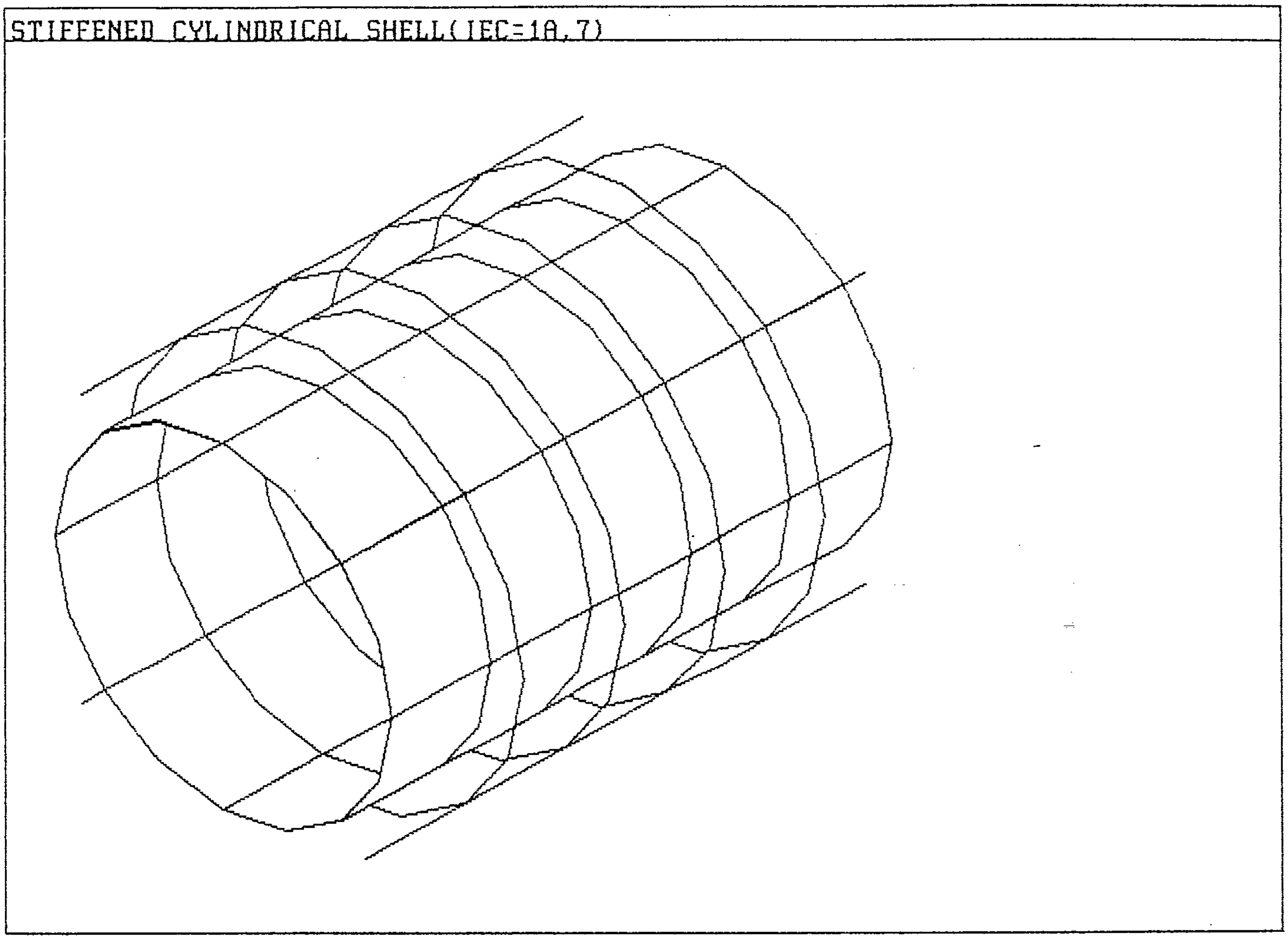


FIGURE 3.1: Stiffened Cylindrical Shell

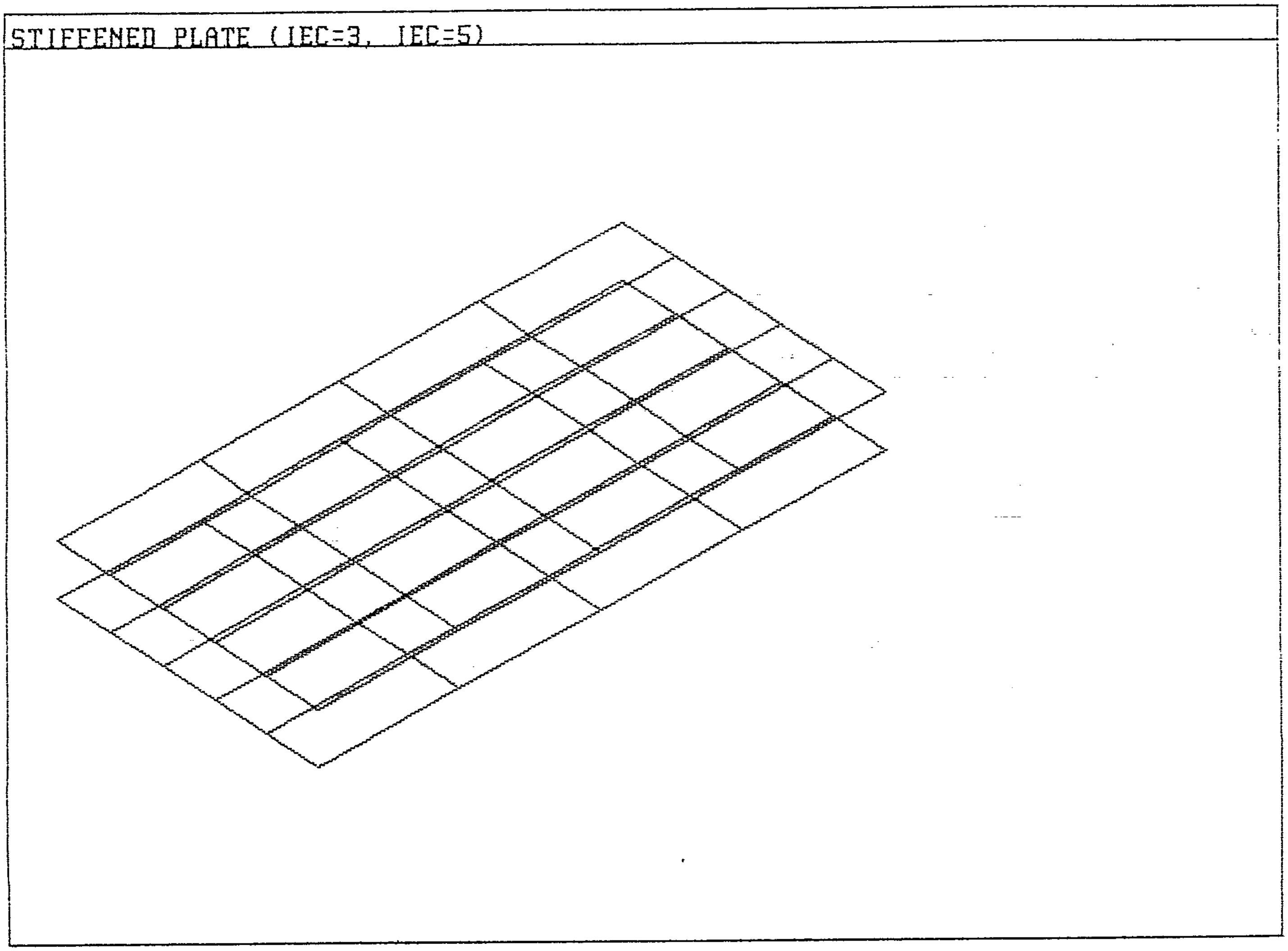


FIGURE 3.2: Stiffened Plate

CHAPTER 4

VERIFICATION OF VASHID PLOTTING CAPABILITIES

The objective of this contract item is to verify the operation of VASHID plotting capabilities for line elements following the implementation of VASMOV enhancements for colour coding by element type, element group, or superelements.

Previously, VASHID plotting capabilities for line elements only included 2-noded bar/beam elements (IEC=3 and 8). At present, the VASHID plotting capabilities for line elements have been extended to include curved or tapered 3-noded bar/beam (IEC=7 and 10).

The illustration of the VASHID plotting capabilities of these four line elements can each be seen in Figure 4-1, Figure 4-2, Figure 4-3, and Figure 4-4 respectively.

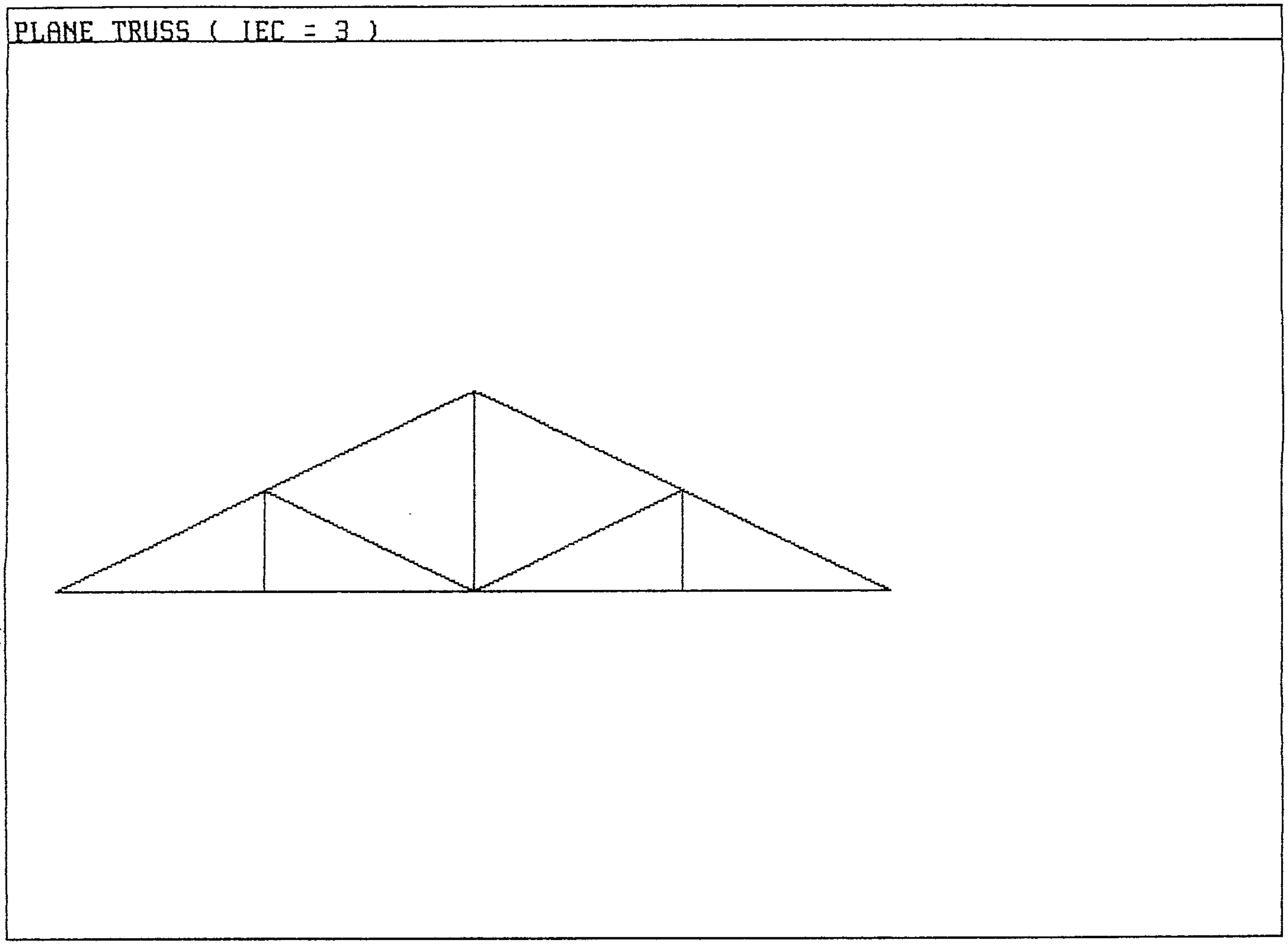


FIGURE 4.1: Plane Truss (IEC=3)

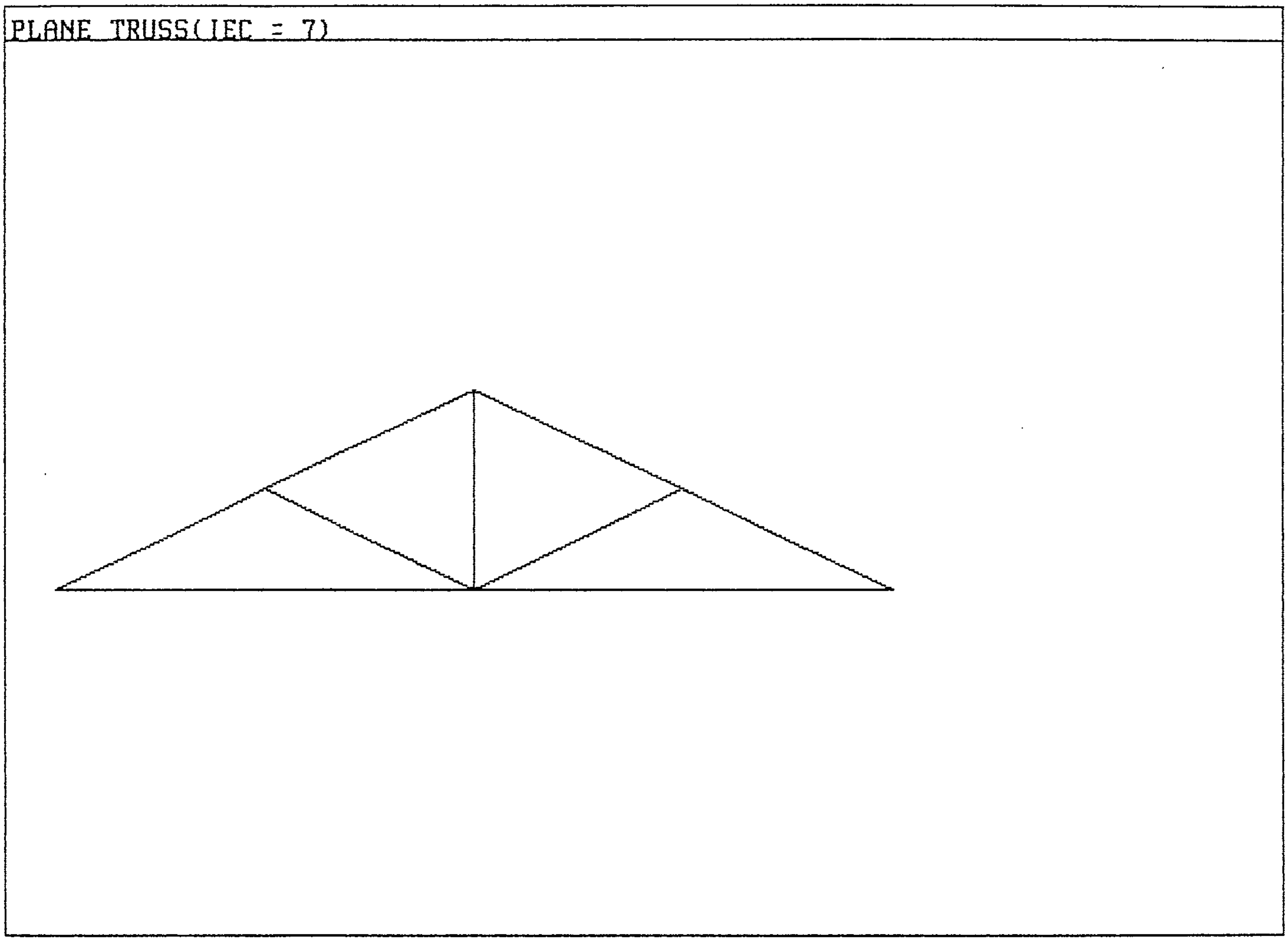


FIGURE 4.2: Plane Truss (IEC=7)

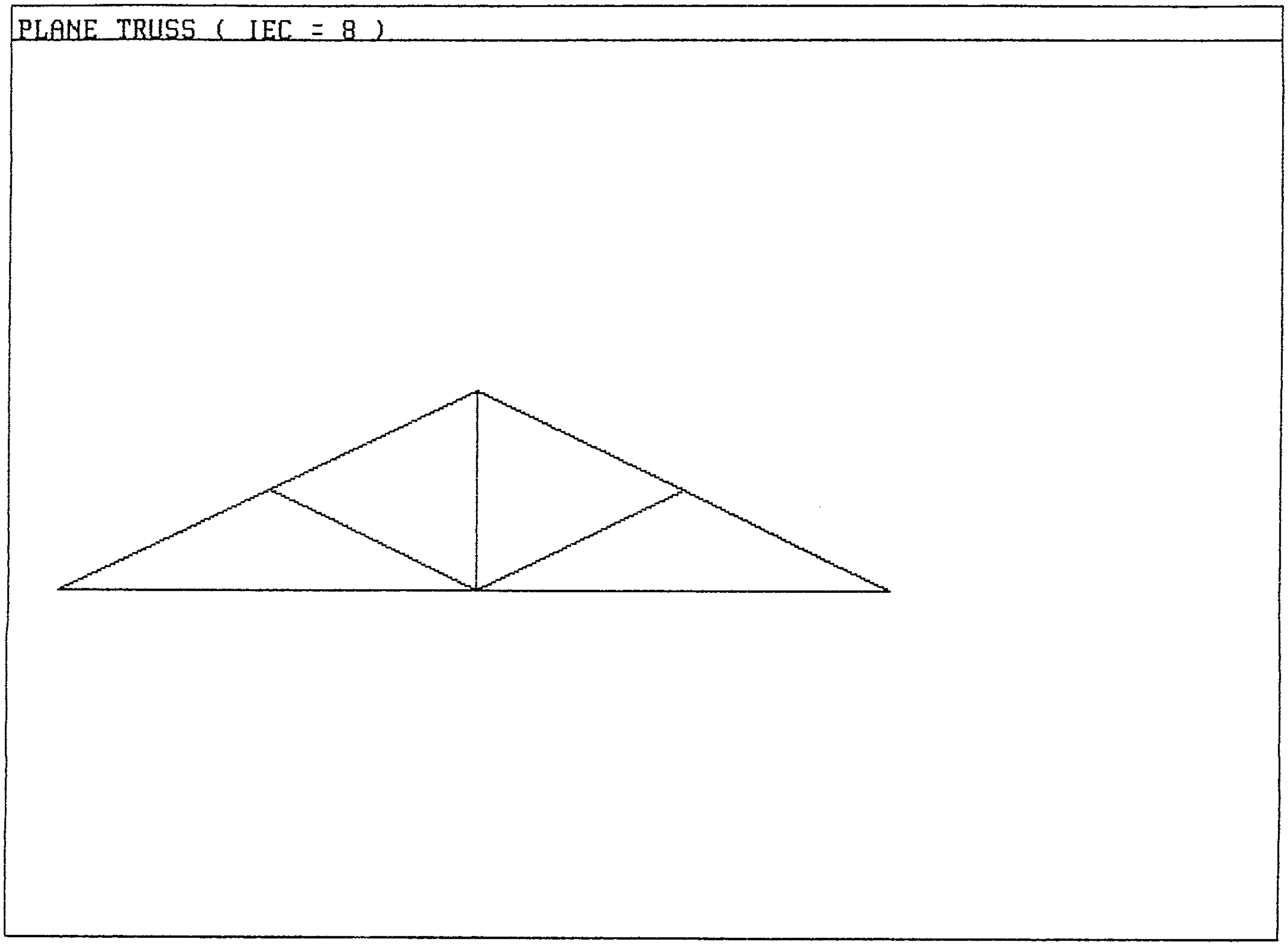


FIGURE 4.3: Plane Truss (IEC=8)

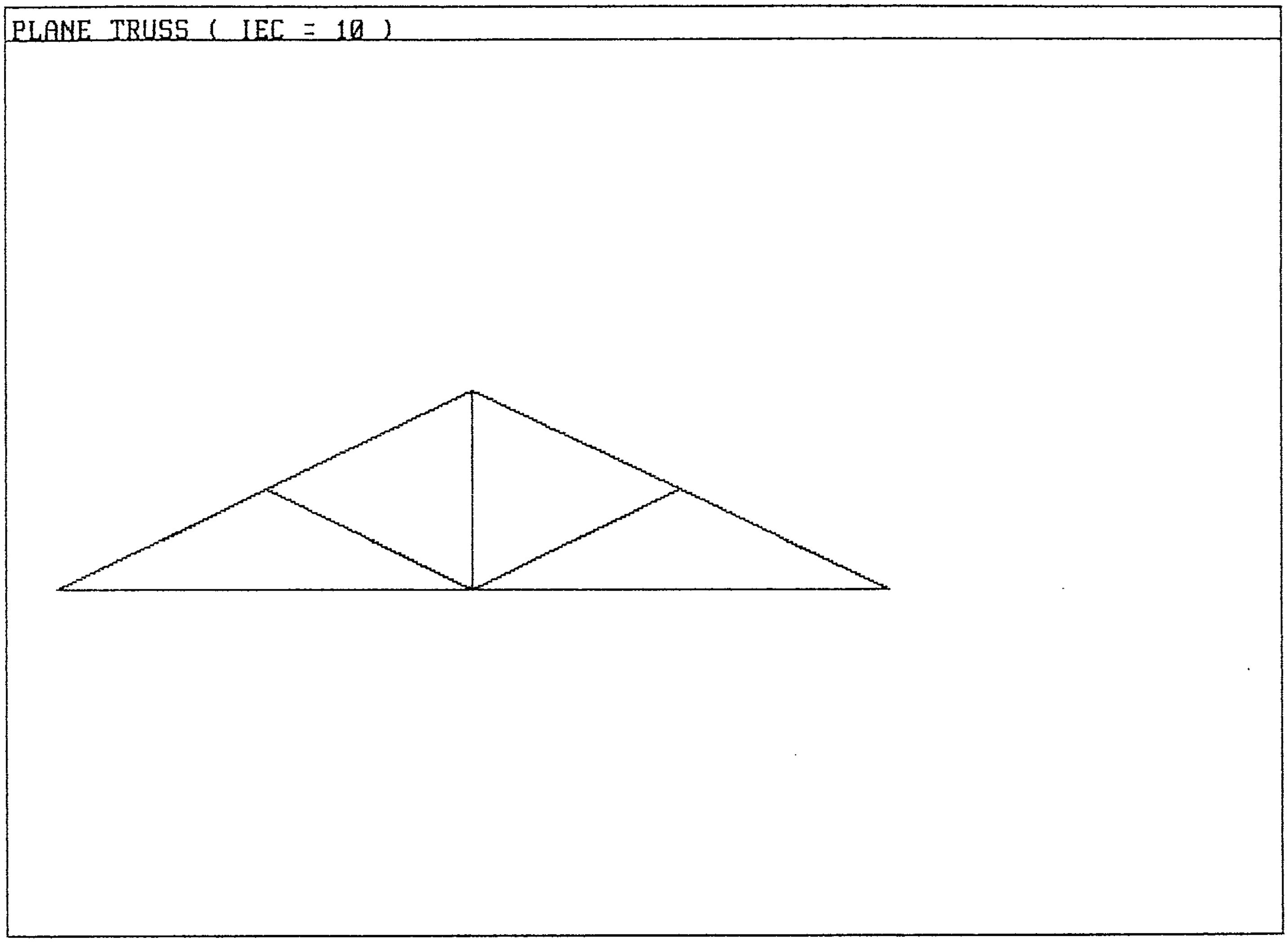


FIGURE 4.4: Plane Truss (IEC=10)

CHAPTER 5

OPTIMIZATION OF ARRAY DIMENSIONS WITHIN VASHID AND INCREASE ITS CAPACITY AND COMPUTATIONAL SPEED

5.1 Introduction

The early development of VASHID was performed as part of a study of potential new hidden line algorithms to avoid the dependence of VASTG on a commercial software for hidden line graphics. The budget for this development was extremely limited and extensive use was therefore made of code from VASMOV which converted finite element connectivity data into panel data and stored it in the IP array. Translation routines were produced to extract the polygon information from the IP array and store it in the LPOLY matrix as required by the hidden line algorithm. A supplementary array called LINEP containing line information was also created. This was relatively straightforward for linear (straight-sided) elements and was performed by subroutine STRAIT. For quadratic elements, the "fine grid" option was automatically activated which had the effect of placing a mesh refined to degree two into the IP array. To accomplish this, subroutine FGRIDH was used. FGRIDH is an extended version of FGRID in VASMOV with appropriate logic to label polygon edges not on edges of the original element as invisible. Subroutine HIDSET then placed polygon information generated by FGRIDH into the LPOLY matrix and edge information into the LINEP array.

The purpose of this contract item has been to optimize array dimensions within VASHID to increase its capacity and to optimize the software coding with regard to computational speed.

5.2 Elimination of XT Array

The XT array with dimension of three times NJMAX was utilized in VASMOV and VASHID of VASTG to define the middle surface nodes for thick/thin shell element of IEC=1 (modelling option 2). The XT array was communicated to the AUXM1 and AUXM2 subroutines of VASSUB module through the common block FSPACE. Closer examination of the above mentioned sections of VASTG code showed the XT array in this respect was redundant.

5.2

It has been found that the information stored in the XT array can be found in the X array (containing the coordinates of the geometric nodes) to define the middle surface nodes. The utilization of midsurface information directly from the X array in both AUXM1 and AUXM2 subroutines has not affected to any extent, the plotting capabilities of deformed models of the thick/thin shell element (IEC=1) of VASTG. The XT array has also been removed from the common block FSPACE to which it was previously attached. As a result, the initial definition of the XT array is no longer required in both VASMOV and VASHID programs and 28,500 storage locations are saved.

5.3 Information on LINEP and LPOLY Arrays

Both line and polygon information are stored in the LINEP and LPOLY arrays, respectively, in a subroutine called HPOLY, which has been newly developed with the objective of speeding up the overall calculation process of VASHID program. Prior to the calling of the subroutine HPOLY, the element connectivity for each polygon and each line is stored in an array called INP of dimension 8, instead of the IP array. In so doing, the IP and hence IP1 arrays are removed entirely from VASHID program, thereby saving 100000 storage locations.

Another significant improvement in terms of computer storage saving can be seen with the elimination of the INVIS array which was used to describe the visibility of edges of polygons. The lines which were invisible were listed in the INVIS array. The dimension of the array was 8000. Every line stored in the LINEP array would be checked against each entry in the INVIS array by subroutine HIDV to determine if it is visible prior to executing the hidden line logic. This check was unnecessarily time consuming.

After considering several alternatives, it was decided that the use of the INVIS array to describe the visibility of every edge of each polygon should be abandoned. Not only could 8000 computer storage locations be saved, but the computational speed could be improved. The method of storing the information related to the visibility of lines which was adopted is quite simple. The terms in the LINEP array defining lines would be made positive or negative depending on whether the lines were visible or invisible. This visibility information can be readily retrieved in subroutine HIDV by checking the sign of the LINEP terms. When the line is negative, all the calculations and checks that follow will be skipped, and next line in LINEP array will be checked against the polygon list. It was found that the computational speed on test problems nearly doubled by simplifiying the visibility check in the manner described.

The storing of lines in LINEP array is performed in subroutine HPOLY but the determination of coded identifiers for lines is in fact performed in subroutine LPACK. Subroutine LPACK will encrypt the two node numbers, L1 and L2, into a single number defining the line between them, using the equation LINE = 10000*LA + LB, where LA and LB are the maximum and minimum of L1 and L2, respectively. This equation is based on the assumption that there are no more than 10000 nodes present. The subroutine LPACK can perform the reverse role of determining the two end nodes, L1 and L2, when a line is given.

5.4 Optimization of Array Dimensions

From the above discussion, it can be seen that nearly 136,000 computer storage locations have been saved. Further storage locations can be saved in the subroutine HCOMPR, which is a modified version of the subroutine COMPR in VASSUB program. VASHID previously called subroutine COMPR but at present calls HCOMPR. These two subroutines are only different in that HCOMPR uses the LINEP array and COMPR uses the IP array. They both serve the same purpose in getting rid of unused nodes from the node list. In COMPR, the IP array is first transferred to NN0 array and NN0 array to NN1 array. These two arrays both occupy 10 times NBMAX(=3001) locations in COMPR, which is extremely inefficient. In HCOMPR, with line information stored in LINEP array, the IP array is transferred to NN0 and NN0 to NN1 arrays. As a result of the use of the LINEP array, the dimensions of NN0 and NN1 in HCOMPR are at present are greatly reduced to only 2 times the maximum dimension of LINEP array. This is another significant storage reduction.

5.5 Computational Speed

The computational speed of hidden line removal algorithm is strongly dependent upon the requirement to check every line to see whether it is visible or invisible and the requirement to check every line against every polygon of which it is not an edge to establish whether the line is above, below, to the right or left, or an edge of, or intersects the polygon. If the line intersects the polygon, the point of intersection will be determined. If the line is behind the polygon, the hidden portion will be determined and stored in the TSTRT and

TEND arrays. All these possible checks and calculations have contributed to the slowing down of the computation of HIDV, the hidden line removal subroutine. These operations must be performed in the most efficient manner possible. The order of these checks has been rearranged so as to speed up the hidden line removal operation. The check of IPOLY, the total number of polygons present, has been introduced in the very beginning. IPOLY has a value of 0 for the case when only line elements are present. Therefore, all other checks will be skipped over if IPOLY of value 0 is encountered. However, the main factor behind the improvement computational speed has come from the fact that NVIS array is used in substitution of INVIS array and hence the check of invisible line stored in INVIS array against the line stored in the LINEP array is eliminated. The check for invisibility of line is simply done by finding out if it is negative, which has been explained in the previous section.

5.6 Comparison of the VASHID Performance Before and After Modifications Made

For the purpose of illustrating the performance of VASHID program after the modifications have been made, a model meshed with 20-noded solid elements (IEC=2) has been generated. Figure 5-1 shows the plotting capability of VASHID before modifications were made whereas Figure 5-2 shows the plotting capability of VASHID after modifications were made. The CPU time of the first plot is approximately 52.21 seconds while that of the second plot is 31.41 seconds. Besides the great improvement in computational speed, a slight change has been made to get rid of the additional lines appearing in the first plot. This has been made in the subroutine PLTSET where the minimums and maximums of each polygon are stored in the DPOLY array. These minimums and maximums have been decreased and increased by 0.0001, respectively.

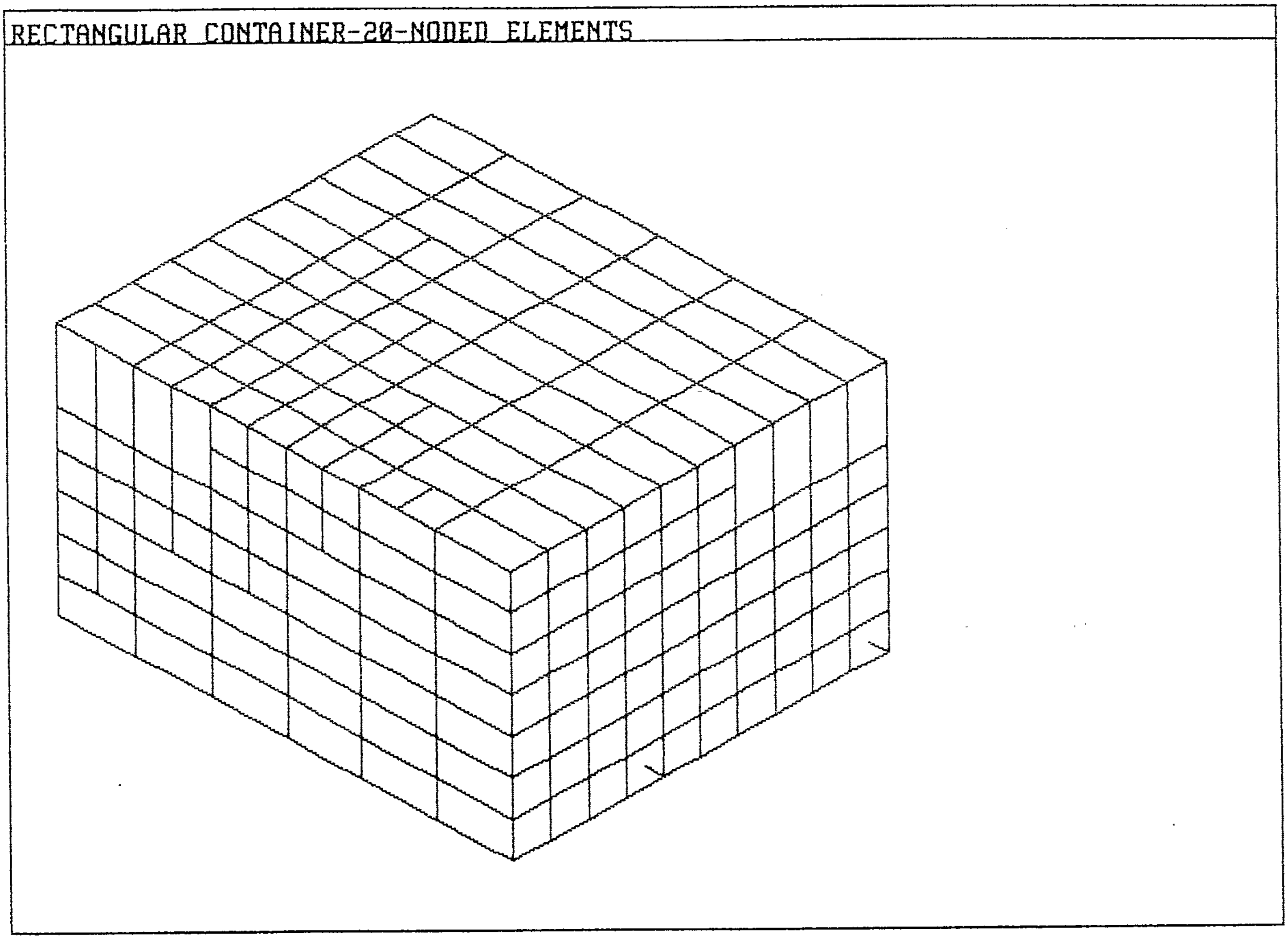


FIGURE 5.1: Performance of VASHID Before Modifications Made
(CPU Time = 52.21 Secs)

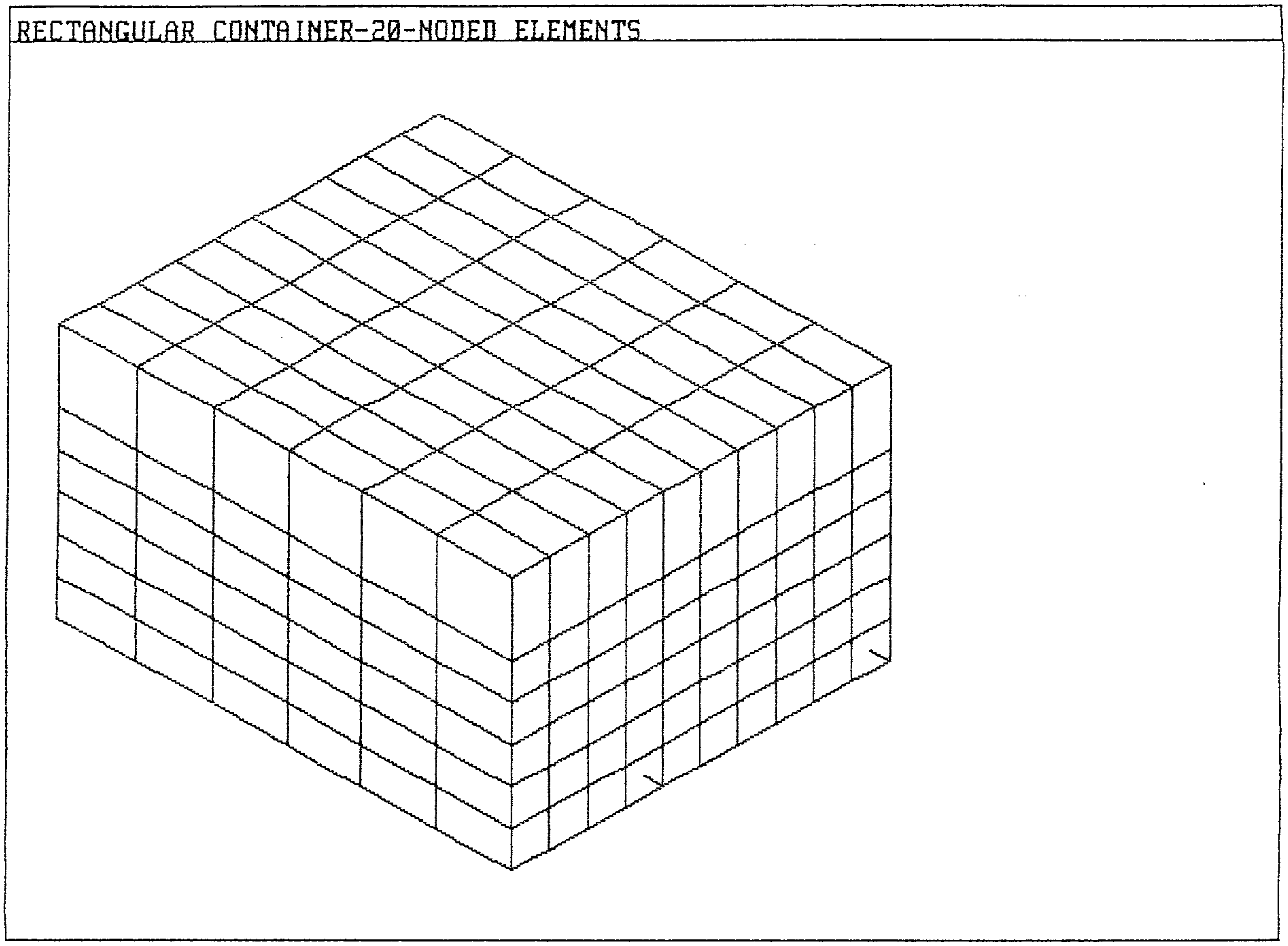


FIGURE 5.2: Performance of VASHID After Modifications Made
(CPU Time = 31.41 Secs)

CHAPTER 6

ENHANCEMENT OF VASTG ELEMENT LOAD PLOTTING CAPABILITIES

6.1 Introduction

The VAST analysis program provides the user with an extensive element library including rods and beams, membranes, plates and shells, and solids. Various types of loads may be applied to the finite element models but basically they can be classified as distributed element loads or concentrated (point) loads. Element loads can be either pressure loads, edge loads, surface loads, line loads, prestrain axial loads, or translational support motion loads. Concentrated loads can take the form of nodal loads or moments. The VASTG graphics package provides the user with the capability to display the load data for verification purposes. The plotting module PLTV12 is the interactive program for the graphical display of the applied loading. It allows the user to plot the loads on the finite element grid of the entire model or selected portions. The user has much control over the data which will be displayed. (See VASTG6 Operating Instructions.) The load plotting capabilities of PLTV12 have been extended under this contract to all possible types of element loadings. The new capabilities which were implemented are discussed in the following sections. While the point load plotting capability was complete, the element load plotting capability had not been fully implemented and this chapter describes work done to address those deficiencies. The types of element loadings that can now be plotted are summarized in Table 6.1.

Translational support motion loads are difficult to plot prior to analysis. Therefor, this type of load data does not yet have any graphical verification capability and the user is informed through a warning message of this limitation when using the VASTG graphics package.

6.2 Edge Pressure Loading

The PLTV12 program previously did not plot element edge loadings which are available for membrane and plate, shell elements listed. A similar deficiency existed for the axisymmetric elements. The capability to plot edge loads was therefore added to the PLTV12 module.

6.2

Element types subjected to pressure loads are membranes, plates and shells along the edges or on the surfaces. For element types with IEC values of 1, 2, 6, 16, 17, or 19, only surface loads are considered.

For element types with IEC values of 9, 11, 12, 13,14, 15, 18, 20, 21 or 23, only edge loads are considered. For element type IEC=4 and IEC=5, both edge and surface loads are considered.

6.2.1 Element Surface Load

In the subroutine LOPLOT in PLTV12 program, pressure loads acting on the non-axisymmetric solids (IEC=1,2,6,16,17) are classified as surface loads. These solids may have 3 to 6 nodes on each surface. The loads applied can be either uniform or non-uniform across the surface. Prior to the plotting of the applied pressure load, the surface normal vector must be determined by calling the subroutine AUXL2.

Two exceptional cases are the triangular plate element (IEC =4) and quadrilateral plate element (IEC=5). Element type IEC=4 has 3 edges (S1, S2, S3) and 1 surface (S4) while element type IEC=5 has 4 edges (S1,S2, S3,S4) and 1 surface (S5). Each of these edges and surfaces can bear loads. For these two types of element, loads acting on S4 for IEC=4 and on S5 for IEC=5 are therefore considered surface loads. The subroutine AUXL1 is called prior to the plotting of the surface loads acting on IEC=4 as it is of a special case as for other triangular membranes such as IEC = 9.

The loads acting on the S1, S2 , S3 for IEC=4 and S1, S2, S3, S4 for IEC =5 are known as edge loads. The edge loads will be discussed in the next section.

6.2.2 Element Edge Load

In this section, the deficiency in the plotting capabilities of edge loads is addressed for the remaining elements in the VAST element library. In the subroutine LOPLOT in PLTV12 program, the loads applied are identified as edge loads or surface loads. As discussed in the previous section, load applied to edges of element types IEC = 4 (S1,S2,S3) and IEC =5 (S1,S2,S3,S4) are examples of edge loads. Similarly, element loads acting on axisymmetric solids and membranes (IEC=12,13,14,15,19) and (IEC=9, 11, 18, 20, 21, 23) are also known as edge loads.

The direction vector of surface normal on which edge load is applied and the direction vector of the edge are required to determine the direction vector of the edge normal. The cross product of the surface normal vector and the edge vector (a new subroutine AUXL3 was developed to perform these calculations) is then used to plot the edge loads applied.

For demonstration purposes, let us consider a rectangular panel meshed with 20 elements of quadrilateral plates (IEC=5). Figure 6.1 shows a typical edge load acting on edge 1 (denoted as S1) when the panel is meshed with 20 quadrilateral plate bending elements (IEC=5). Similarly, Figure 6.2 shows an edge load on edge 4 (S4) when the panel is meshed with shear web elements.

6.3 Element Point Loads

The capability to plot element point loads has been implemented in PLOTV12. The VAST element types capable of this type of loading are listed in Table 6.3.

The completion of this contract item required modifications to the MARLIB library subroutine LOAD1 which generates the standard scratch file of model loads. Modifications were required to store the point load data for all element types, previously this data was skip read over. The PLOTV12 program was restructured to include a new section of code to plot the point loads.

6.4 Rod and Beam Element Loading

The rod and beam elements are capable of being loaded by distributed loading (line loads), axial prestrain and point loads. Under this contract, the capability to plot the distributed loading for the general beam elements (IEC=3), the curved beam (IEC=7) and the axisymmetric beam element (IEC=22) has been added. The capability to plot the axial pre-strain for the beam element (IEC=3) and bar element (IEC=8) has been implemented.

6.4

6.4.1 Distributed Loads

Distributed loads, which are uniform (if IDISTRL = 1) or non-uniform (IDISTRL = 2), can be applied to beam and rod elements (IEC = 3 and 7). The loads can be orientated in the local y-direction, z-direction or both directions. The subroutine DIST is called prior to the calling of the new subroutine BEAMLO which is used to plot load vectors particularly for beam and rod elements.

Beam element type (IEC=3) is used as an example in the plotting capabilities of distributed loads. The plotting capabilities of the distributed loads on this element is illustrated in Figure 6.3.

6.4.2 Element Point Load

The beam and rod elements can often be subjected to point loads. These point loads can be applied, at various locations along but perpendicular to their axes either in y- or z-direction or both. The location at which a point load is applied is dictated by the local coordinate system of the beam or rod element, in terms of XL, where XL is in the range of 0 and 1 for beam element IEC=3 and in the range of -1 and +1 for beam element IEC=7. As many point loads as desired can be applied to the beam element, the program is terminated when the indicator IPL of 0 is encountered.

Suppose point loads are applied at locations XL =0.0, 0.2,0.4,0.6,0.8,1.0 along the beam element type IEC =3. The plotting capabilities for these point loads are demonstrated in Figure 6.4.

6.4.3 Equivalent Load Due To Prestrain Condition

Besides distributed and point loads, an axial load can also be applied due to the prestrain condition. This prestrain load is considered for the beam elements (IEC=3) and bar elements (IEC=8). For the bar element, the only applied loading is equivalent axial loads due to prestrain condition.

Figure 6.5 illustrates the plotting capabilities of computed equivalent loads due to the prestrain axial load on the beam element IEC=3. The axial loads due to the prestrain conditions are differentiated from beam element distributed load and point load with an addition of a cross sign.

The axial loads due to the pre-strain conditions are differentiated from beam element distributed load and point data through the display of a cross. It must be noted that the axial load vectors do not represent their actual magnitude. The capability to plot point loads for the beam/rod elements was developed in such a manner that other elements having point loading capability would use the same algorithm. This feature is described in the following section.

Table 6.1
Applied Loading - Membranes

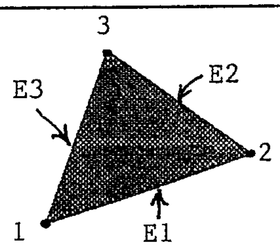
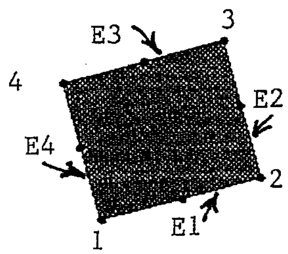
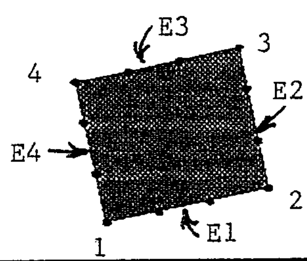
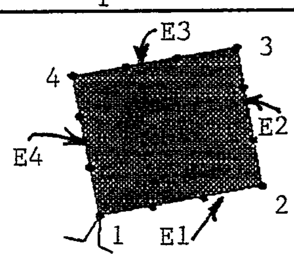
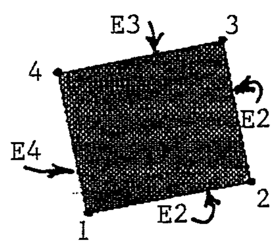
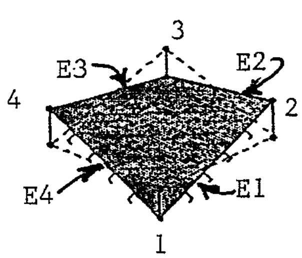
Element	Description	Element Code (IEC)	Number of Nodes	Point Load-ing	Notes	Pressure Loading					
	Triangular membrane (constant strain)	9	3	No	Edge loading only	E1	E2	E2			
	Quadrilateral membr-ane (can be degener-ated into triangle which is identified by repeated node num-ber for 3/4)	20	8	No	Edge loading only	E1	E2	E3	E4		
	Quadrilateral memb-rane	18	12	No	Edge loading only	E1	E2	E3	E4		
	Fracture (K_I and K_{II} are degrees-of-free-dom at node 13)	21	13	No	Edge loading only	E1	E2	E3	E4		
	Shear web	23	4	No	Edge loading only	E1	E2	E3	E4		
	Warped/stiffened mem-brane	11	4	No	Edge loading only	E1	E2	E3	E4		

Table 6.1
Applied Loading - Plates and Shells

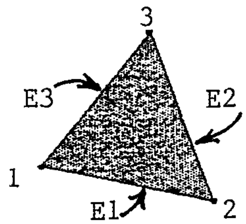
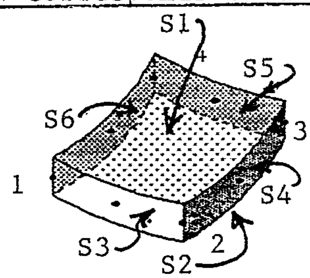
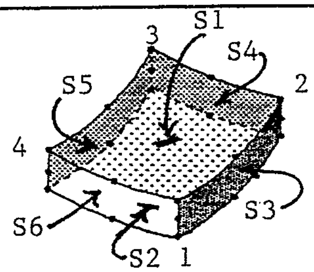
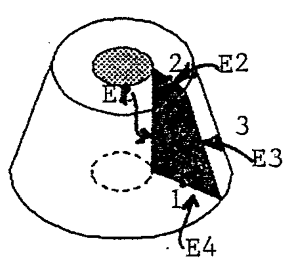
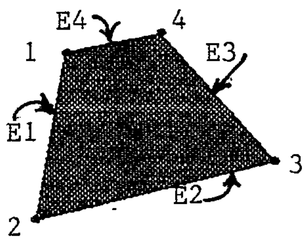
Element	Description	Element Code (IEC)	Number of Nodes	Point Load-ing	Notes	Pressure Loading					
 S4 corresponds to midplane	Triangular plate (membrane and bend-ing)	4	3	No	Edge and mid plane normal press-ure	E1	E2	E3	S4		
	Thick/thin shell	1	8	Yes	Surface top and bottom and edge loading	S1	S2	E1	E2	E3	E4
	Transition (to join thick/thin shell to solid brick)	6	13	Yes	All sur-faces	S1	S2	S3	S4	S5	S6
	Axisymmetric thick/thin shell	19	3	No	Edge loading only	E1	E2	E3	E4		
 S5 corresponds to midplane	Quadrilateral shell	5	4	No	Edge and mid plane pres-sure	E1	E2	E3	E4	S5	

Table 6.1
Applied Loading - Solids

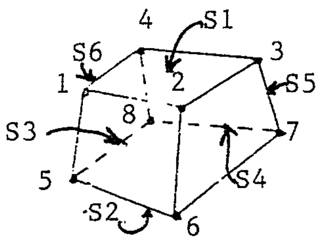
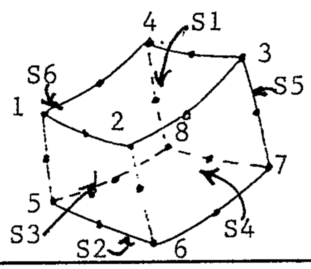
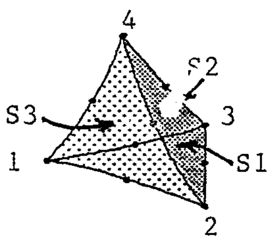
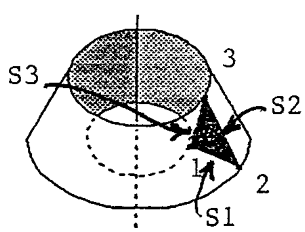
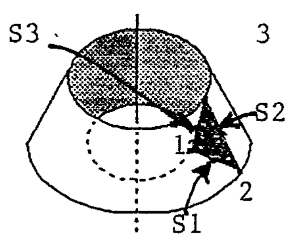
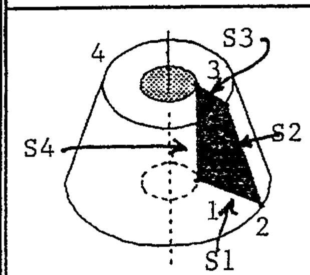
Element	Description	Element Code (IEC)	Number of Nodes	Point Loading	Notes	Pressure Loading					
	Brick (can be degenerated into wedge collapse on surface 4)	16	8	Yes	S4 not applicable to degenerate geometry	S1	S2	S3	S4	S5	S6
	Brick (can be degenerated into wedge collapse on surface 4)	2	20	Yes	S4 not applicable to degenerate geometry	S1	S2	S3	S4	S5	S6
	Tetrahedron	17	10	Yes	Only '3' surfaces to be loaded	S1	S2	S3			
	Axisymmetric triangle	12	3	No		S1	S2	S3			
	Axisymmetric triangle	13	6	No		S1	S2	S3			
	Axisymmetric quadrilateral	14	4	No		S1	S2	S3	S4		

Table 6.1 - Continued
Applied Loading - Solids

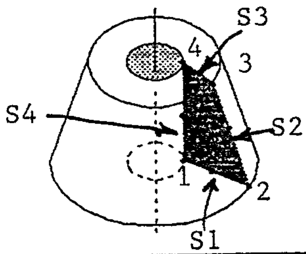
Element	Description	Element Code (IEC)	Number of Nodes	Point Load-ing	Notes	Pressure Loading					
	Axisymmetric quadri-lateral	15	8	No		S1	S2	S3	S4		

TABLE 6.2: VAST Elements for Which Element Pressure Load Plotting Capability was Implemented in PLOTV12

VAST ELEMENT CODE	DESCRIPTION	ELEMENT LOADING
9	Triangular membrane (constant strain)	Edge loading only
20	Quadrilateral membrane (also triangular)	Edge loading only
18	Quadrilateral membrane	Edge loading only
21	Fracture	Edge loading only
23	Shear web	Edge loading only
11	Warped/stiffened membrane	Edge loading only
4	Triangular plate	Edge and mid plane normal
5	Quadrilateral shell	Edge and mid plane normal

TABLE 6.3: VAST Elements for Which Element Point Load Plotting Capability was Implemented in PLOTV12

VAST ELEMENT CODE	DESCRIPTION
3	General beam element
7	Curved beam (for use with thick/thin shell)
1	Thick/thin shell
6	Transition element (join shell to solid)
16	Brick (8-noded)
2	Brick (20-noded)
17	Tetrahedron

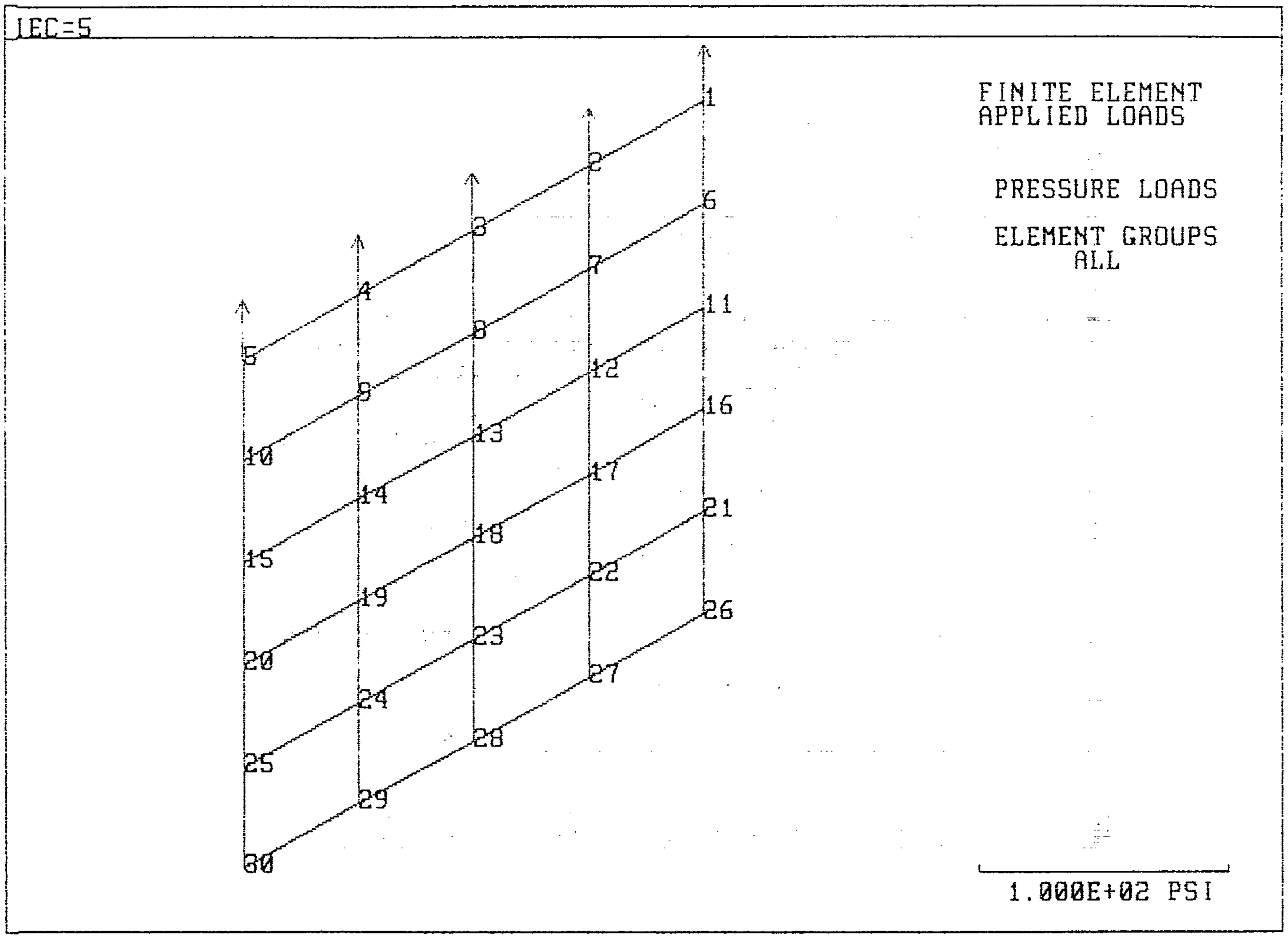


FIGURE 6.1: Edge Loads on Edge 1 (S1) of Quadrilateral Shell Elements (IEC=5)

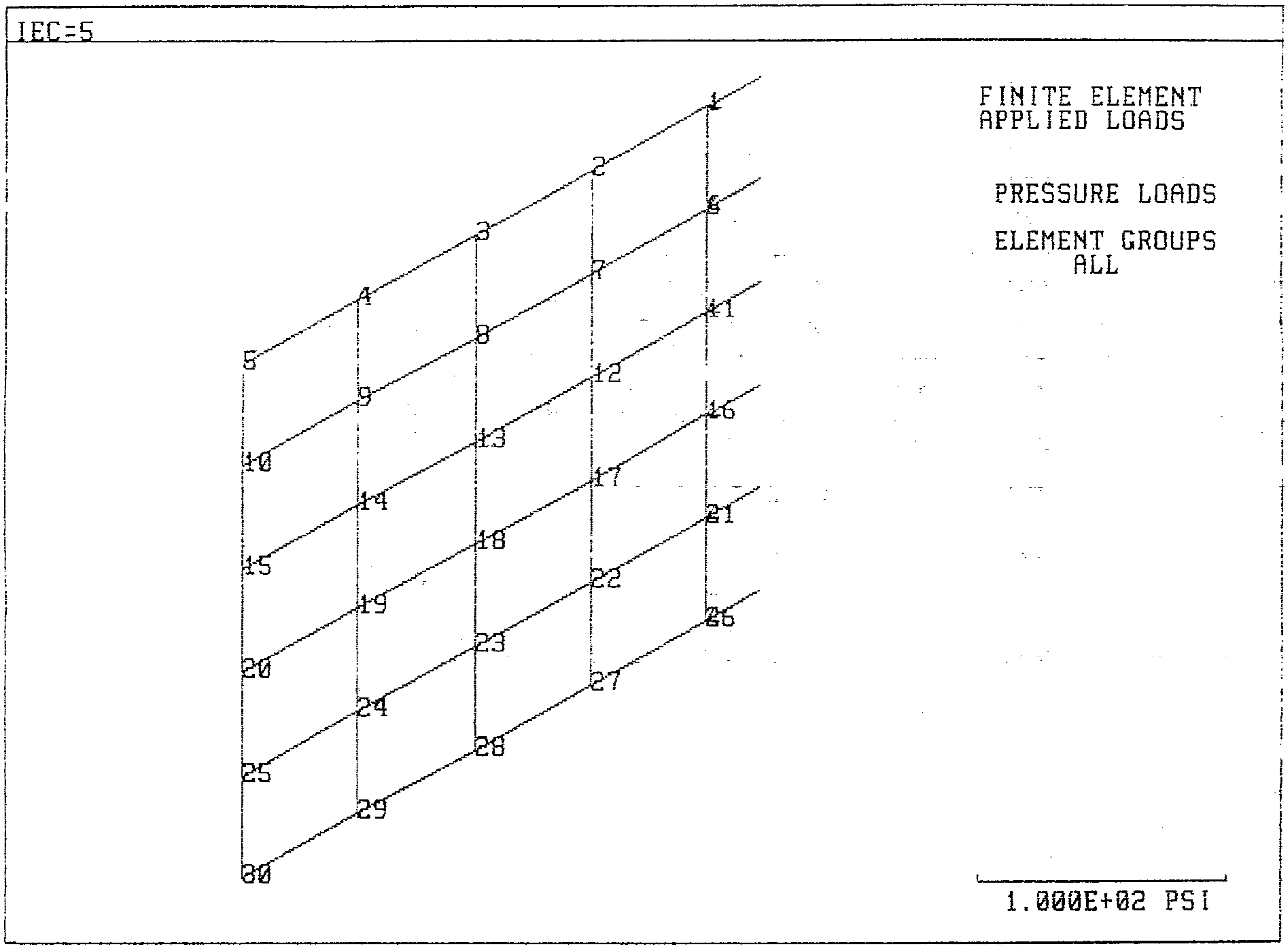


FIGURE 6.2: Edge Loads on Edge 5 (S4) of Shear Web Elements (IEC=23)

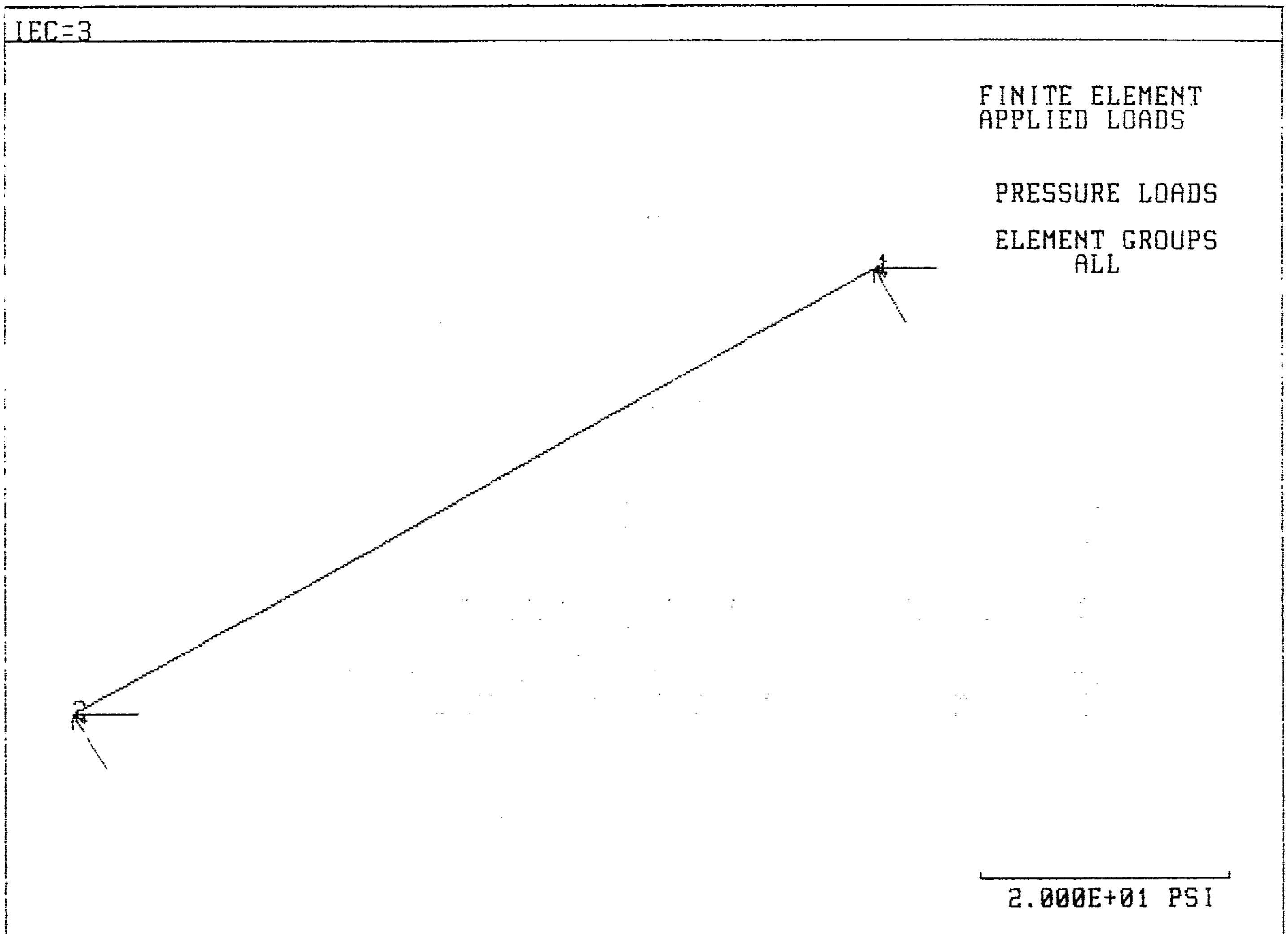


FIGURE 6.3: Distributed Pressure Load Applied on General Beam Element (IEC)

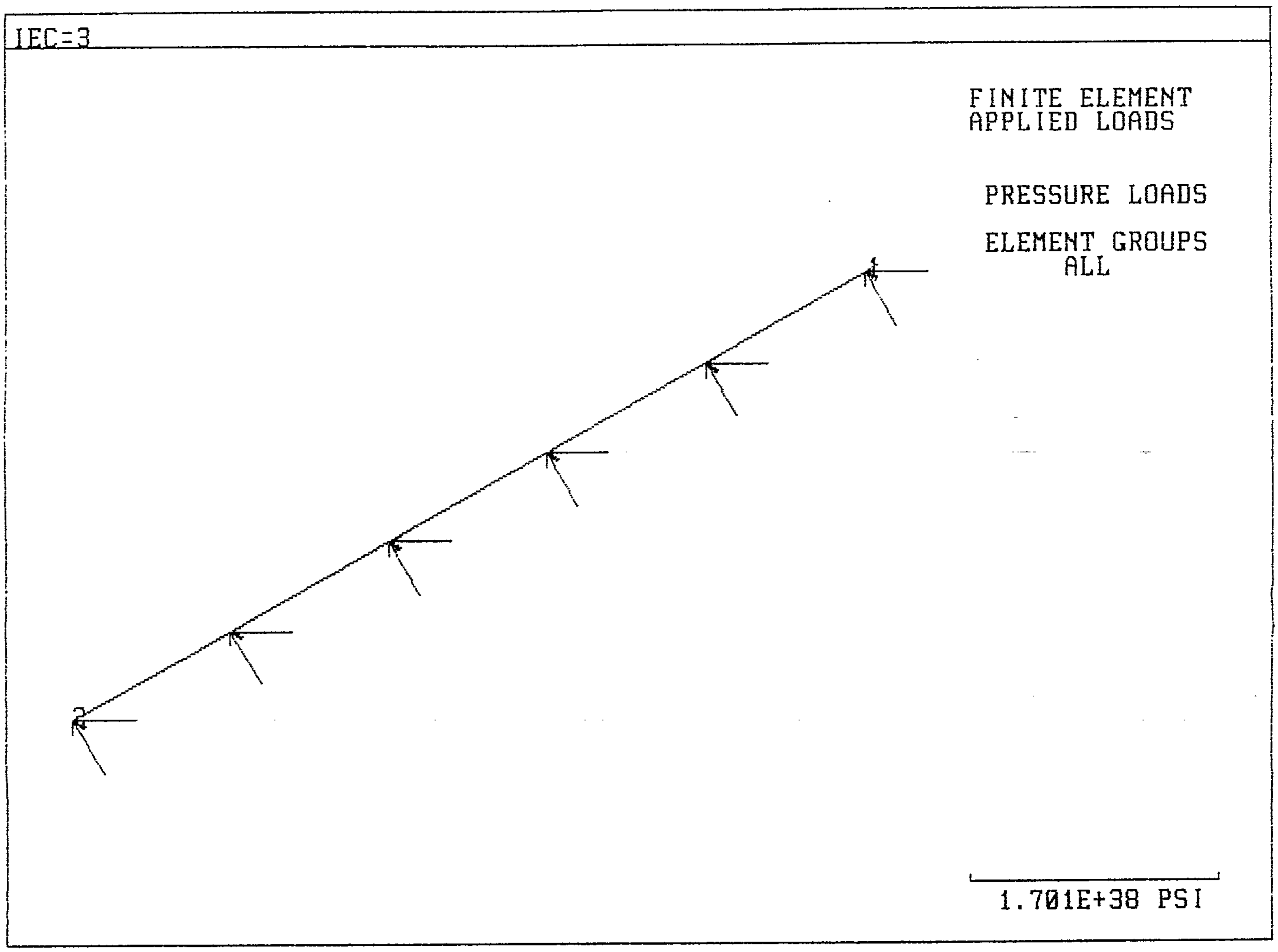


FIGURE 6.4: Point Loads Applied at Six Equally Spaced Locations along the Length of a General Beam Element (IEC=3)

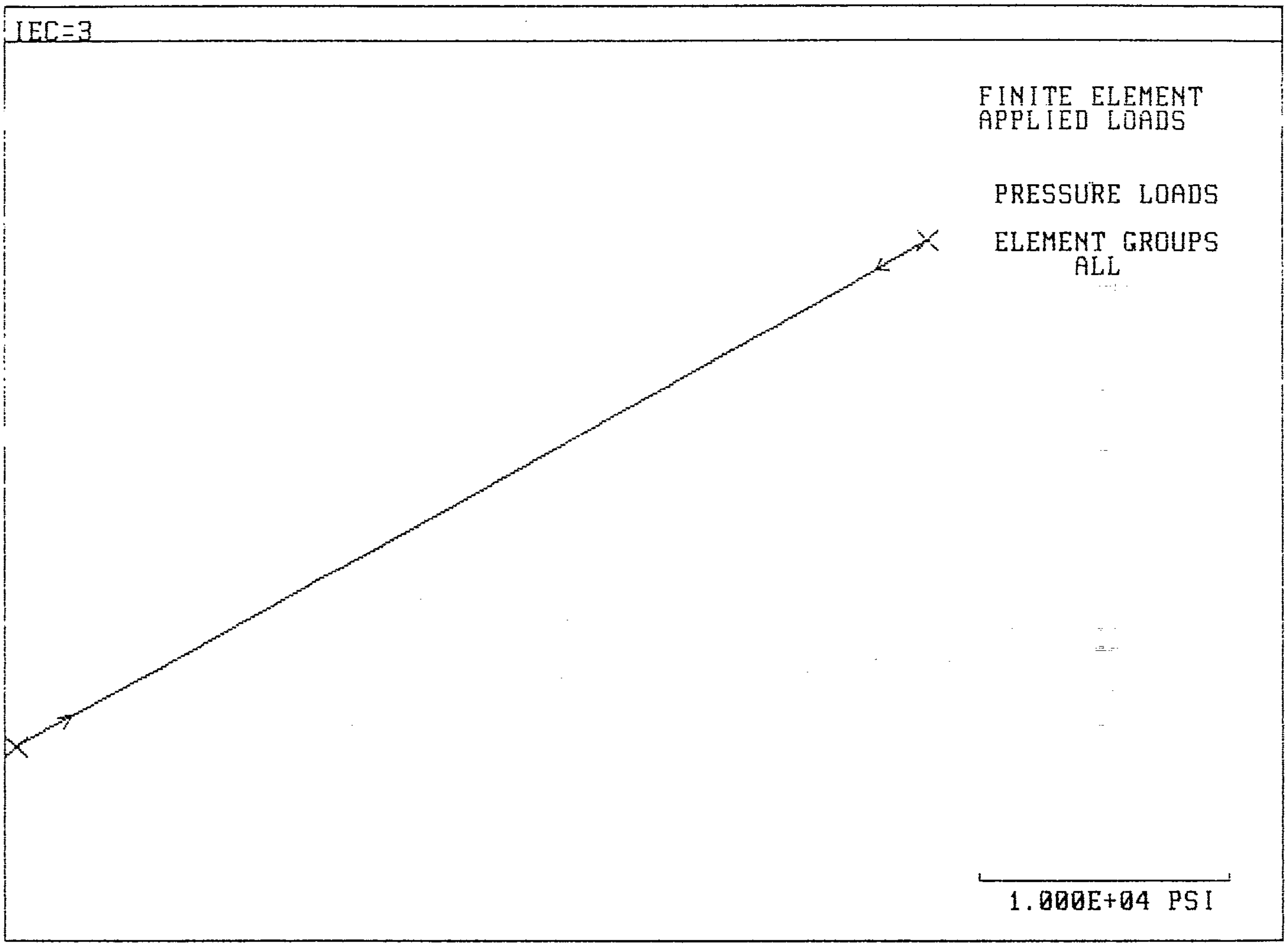


FIGURE 6.5: Prestrain Axial Load For General Beam Element (IEC=3). Symbol "x" Distinguishes This Load from Point Loads

CHAPTER 7

CAPABILITY TO CONTOUR DISPLACEMENTS OR MODE SHAPES ON AUTOMATICALLY IDENTIFIED VISIBLE SURFACES

7.1 Introduction

The capability to contour displacements or mode shapes on automatically identified visible surfaces has been implemented into VASTG. This capability is extremely valuable for a user who has utilized a model generating program to generate his model otherwise or to a user who is unfamiliar with the element and/or surface numbering scheme of his model. Moreover, even for a user who, for other reasons, is familiar with the element/surface numbering scheme of his model, this capability eliminates the tedious prompting previously required to generate a plot in which the visible surfaces were determined by the user. The implementation of this capability and sample problem are given in the following sections.

7.2 Implementation

The PLOTV9 plotting module displays the contours of displacements or mode shapes on the element surface plotting specifications defined by ELOPT. Recall that subroutine ELOPT of the PLOTV library prompts the user for the element surface plotting specifications in standard form and fills the IOPT2 array. The format of the IOPT2 array is given in Table 7.1 and the VAST element types requiring surface specifications as identified by the IOPT2 array are listed in Table 7.2. The required capability to plot the contours of displacements or made shapes on automatically identified visible surfaces involves the definition of the IOPT2 array appropriately for any user selected view using basic hidden line concepts.

Some components of required capability were developed in an earlier contract involving enhancements of VASTG [4] to plot stress contours on automatically identified visible surfaces. The VASMOV program converts the geometry of the model into panels or polygons and removes the surfaces that are internal to the model. Each of the external panels was identified with a "packed" number which stores the superelement number, group number, element number, and the surface number. The packed form of the data identifying external surfaces was stored in the PREFX.VIS file. A new module, POSVIS, was developed to process the PREFX.VIS filed generated by the translator program VASMOV to create a PREFX.P25 file. The POSVIS module utilizes the Poorman algorithm to eliminate surfaces that have backward orientated normals. The revised list of surfaces were used to plot stress contours with in the PLOTV3 module.

A similar algorithm was required within PLOTV9 to identify the visible surfaces automatically. The implementation of the POSVIS module in conjunction with displacement contour plotting with PLOTV9 required some major changes and testing of the module. Some aspects of the algorithms utilized in POSVIS for stress contouring plotting lacked sufficient generality to be utilized directly in PLOTV9. These were addressed first. Of course, the VASTG6 driver module was also modified to provide the user in the option of using automatic visible surface definition. If this option was selected, the driver program directed control to the POSVIS module to generate the PREFX.P25 file.

The PLOTV9 module was modified to indicate that the option of using automatic visible surface definition had been selected. The user cannot change the view or element surface plotting specifications without exiting the plotting module and re-running the POSVIS module.

7.3 Sample Problem

The sample problem selected to illustrate the new capability to display the displacement contours on automatically identified visible surface was a cantilevered beam with concentrated end loads. This sample problem demonstrates the successful implementation of this capability within PLOTV9.

The VAST input data for this model was generated using the VASFEM suite of programs. The VASGEN program module was used to generate the PREFX.GOM file containing the model geometry which consists of ten 20-noded solid brick elements. The capability to display the model using the PLOTV1 program, called by VASGEN, was used and the resulting plot is shown in Figure 7-1. The VASTBC program module was used to generate the PREFX.SMD file containing the boundary condition data. Figure 7-2 was generated using the VASTBC program. The nodes located on the end of the beam were windowed using the 'M' (multiple node) option and then assigned stiffness codes on all degrees-of-freedom. The VASLOD program module was used to generate the PREFX.LOD file containing the load data. The nodes on the end of the beam were individually identified using the 'I' (individual node) option and assigned a concentrated load in the z direction. Figure 7-3 is a plot of the loading condition generated by the PLTV12 program called by VASLOD. The VASUSE program module was used to generate the PREFX.USE file which is the main input data set file for the VAST program. The above process for model generation in VAST is frequently used and does not require the analyst to become familiar with node numbering, element numbering, or element orientation.

The VAST analysis program was then executed to perform the static analysis of the cantilever model. Finally, the VASTG6 program was executed to display the displacement contours on the model using the PLOTV9 plotting module. The feature to define visible surfaces automatically was selected. Figure 7-4 is a plot of the model only produced by PLOTV9. The reader will note that only the model boundaries are displayed (individual elements are not shown). Figure 7-5 is a plot of the displacement contours generated by entering the command directive 'PLOT' and using the preset default responses. Figure 7-6 was generated by a window in the middle portion of the model. As is seen readily, the contours are successfully displayed on visible surfaces only.

TABLE 7.1: Format of Surface Plotting Specifications File

	NEGT	1-st Record
	IOPT(I,1), I=1,12	2-nd Record
	IOPT(I,2), I=1,12	3-rd Record
	*	
	*	
	*	
	IOPT(I,NEGT),I=1,12	(NEGT+1)-th Record
where		
	NEGT is the total number of element groups in the plotting specifications (which might be different from the number of element groups in the finite element model)	
	IOPT(1,I) is the superelement # (equal to zero if NSS is zero)	
	IOPT(2,I) is the element group number	
	IOPT(3,I) is the element type	
	IOPT(4,I) is the number of elements in this group	
	IOPT(5,I) is the first element to be plotted	
	IOPT(6,I) is the last element to be plotted	
	IOPT(7,I) to IOPT(12,I) contain the surface specifications, if required (0 for no plotting: 1 for plotting)	

TABLE 7.2: VAST Element Types Requiring Surface Specifications

VAST ELEMENT CODE	DESCRIPTION	NUMBER OF SURFACES
1	Thick/thin shell	6
2	20-noded brick	6
6	Transition element	6
16	8-noded brick	6
17	Tetrahedron	4

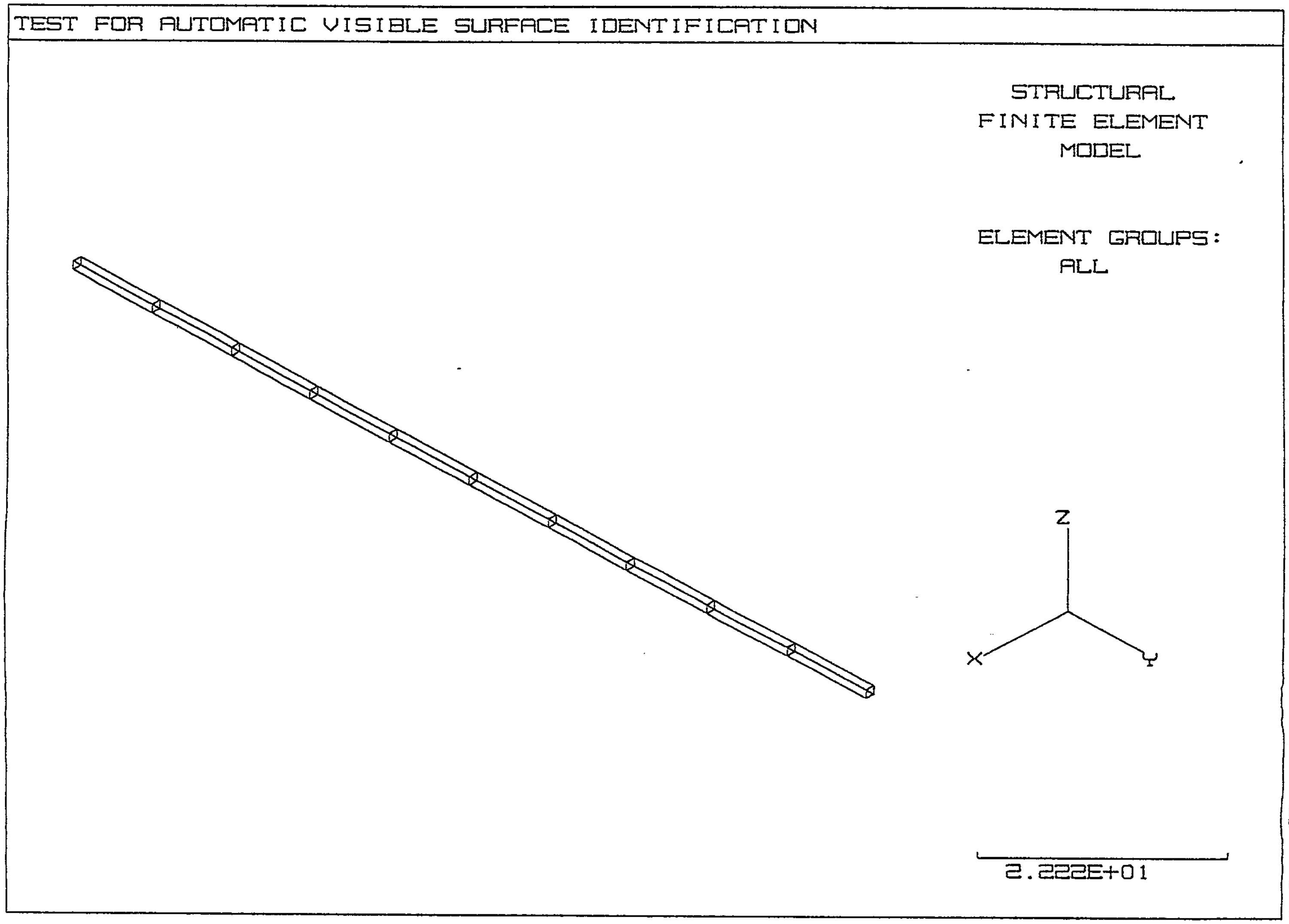


FIGURE 7.1: Sample Problem F.E. Model Plot

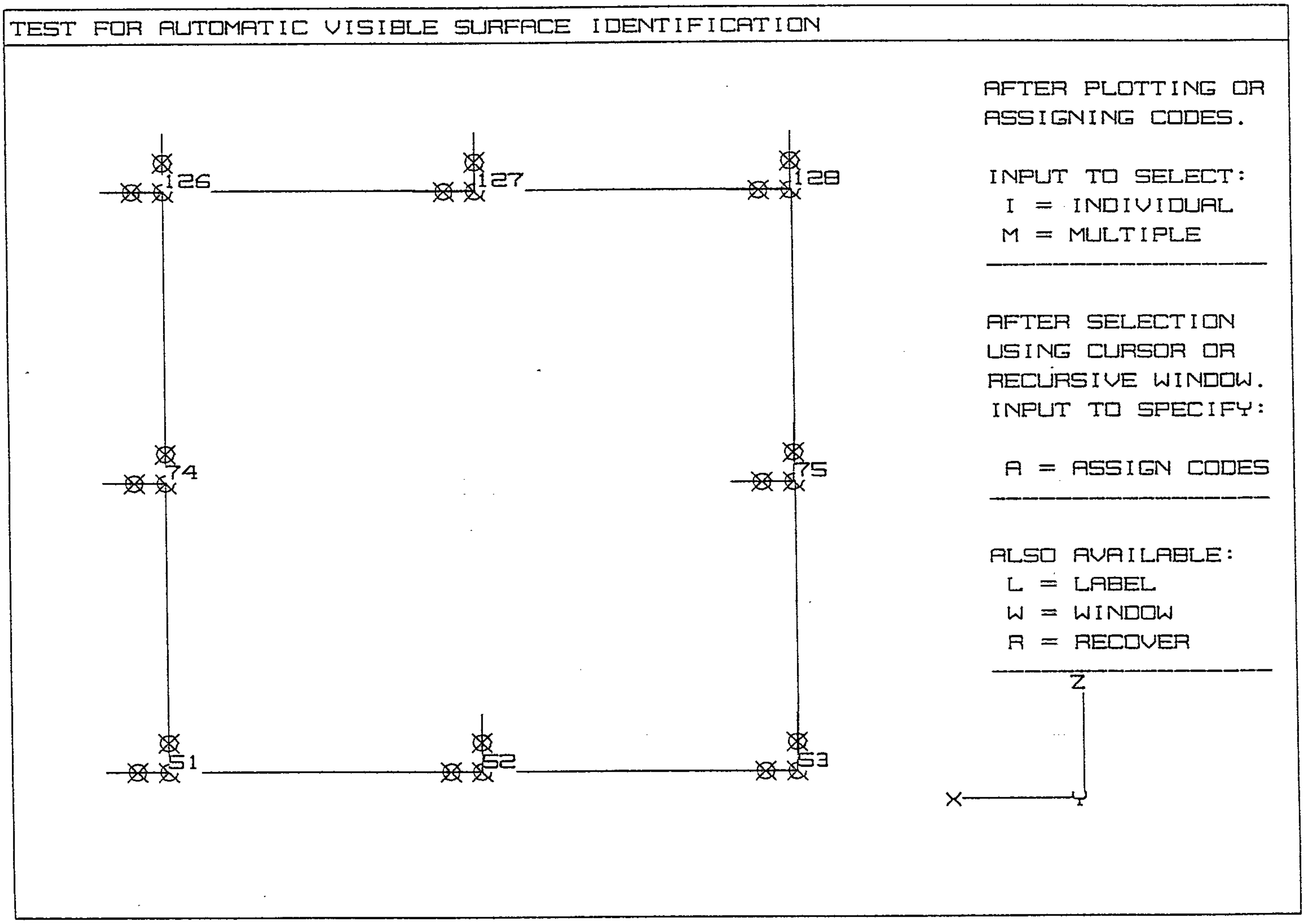


FIGURE 7.2: VASTBC Plot of Boundary Conditions

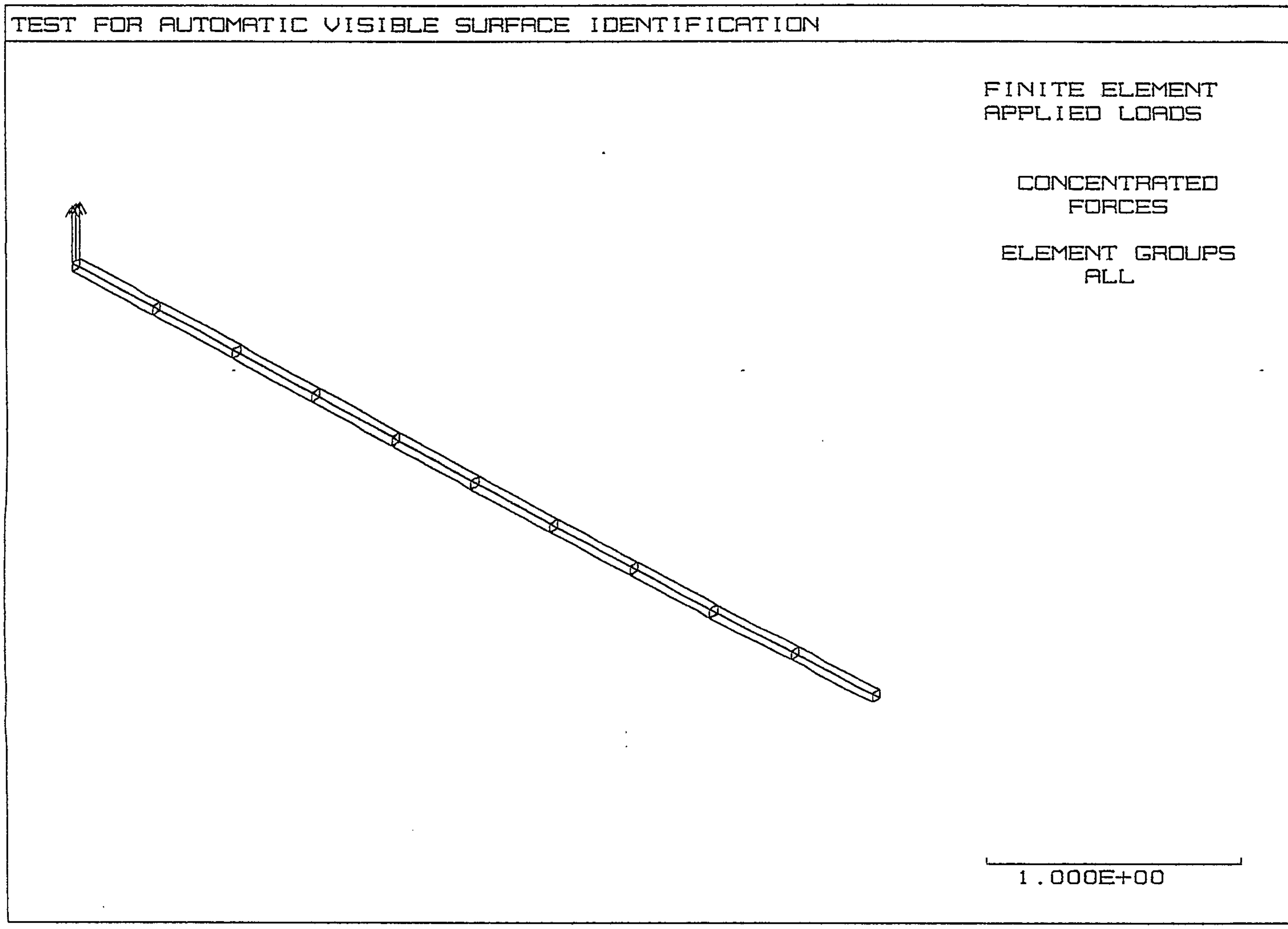


FIGURE 7.3: Concentrated Loads

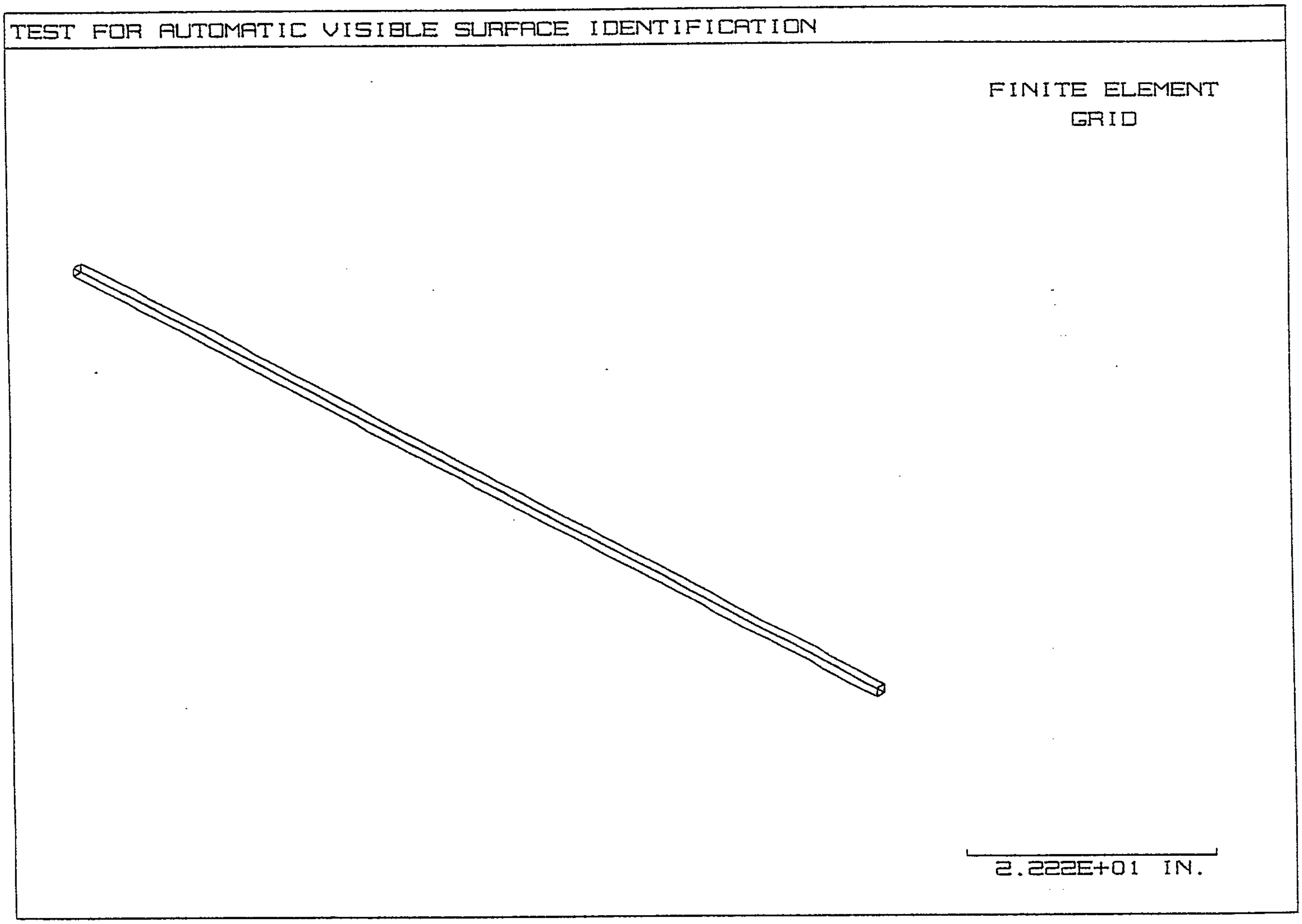


FIGURE 7.4: Visible Surfaces Defined Automatically

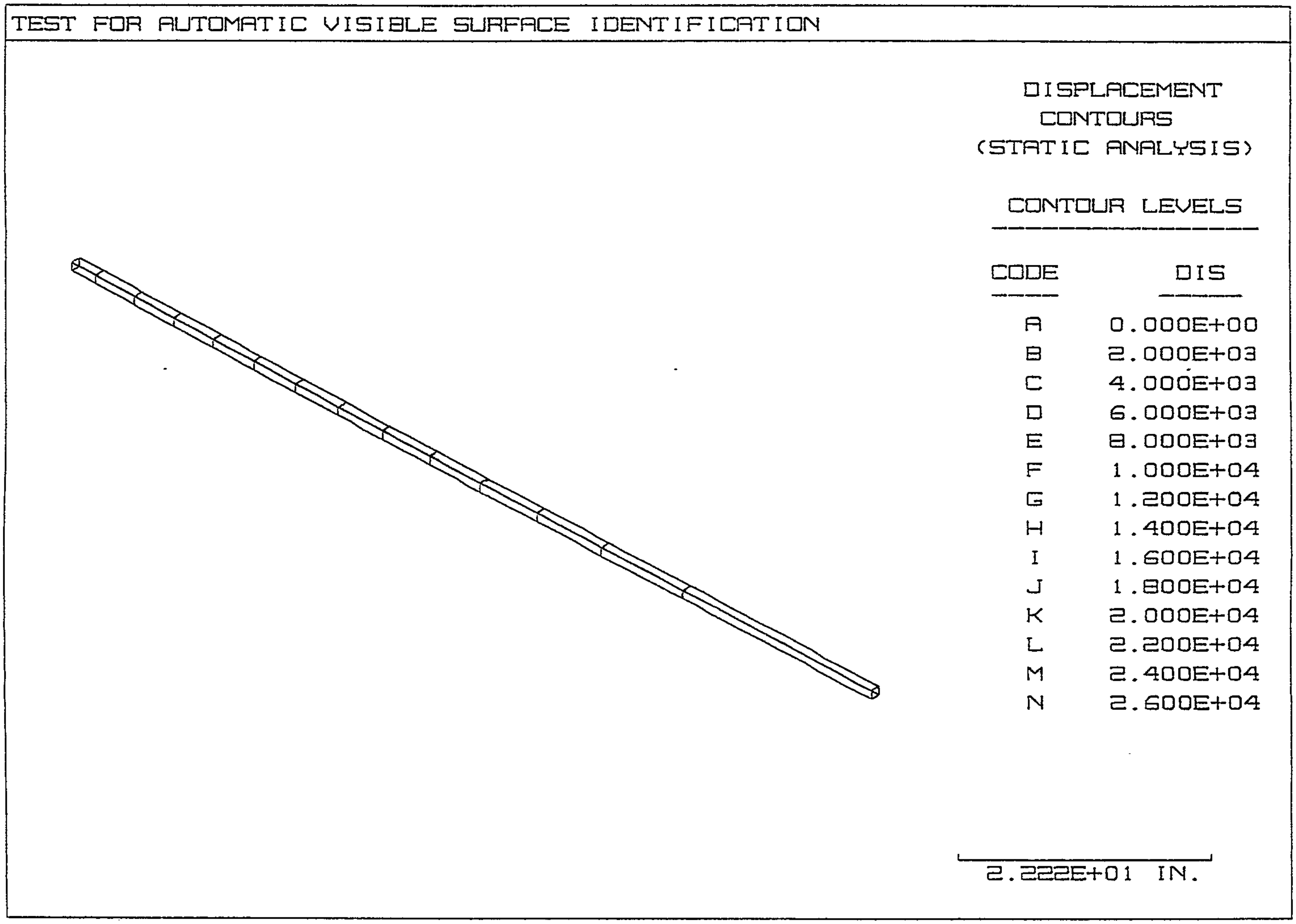


FIGURE 7.5: Displacement Contours on Visible Surface

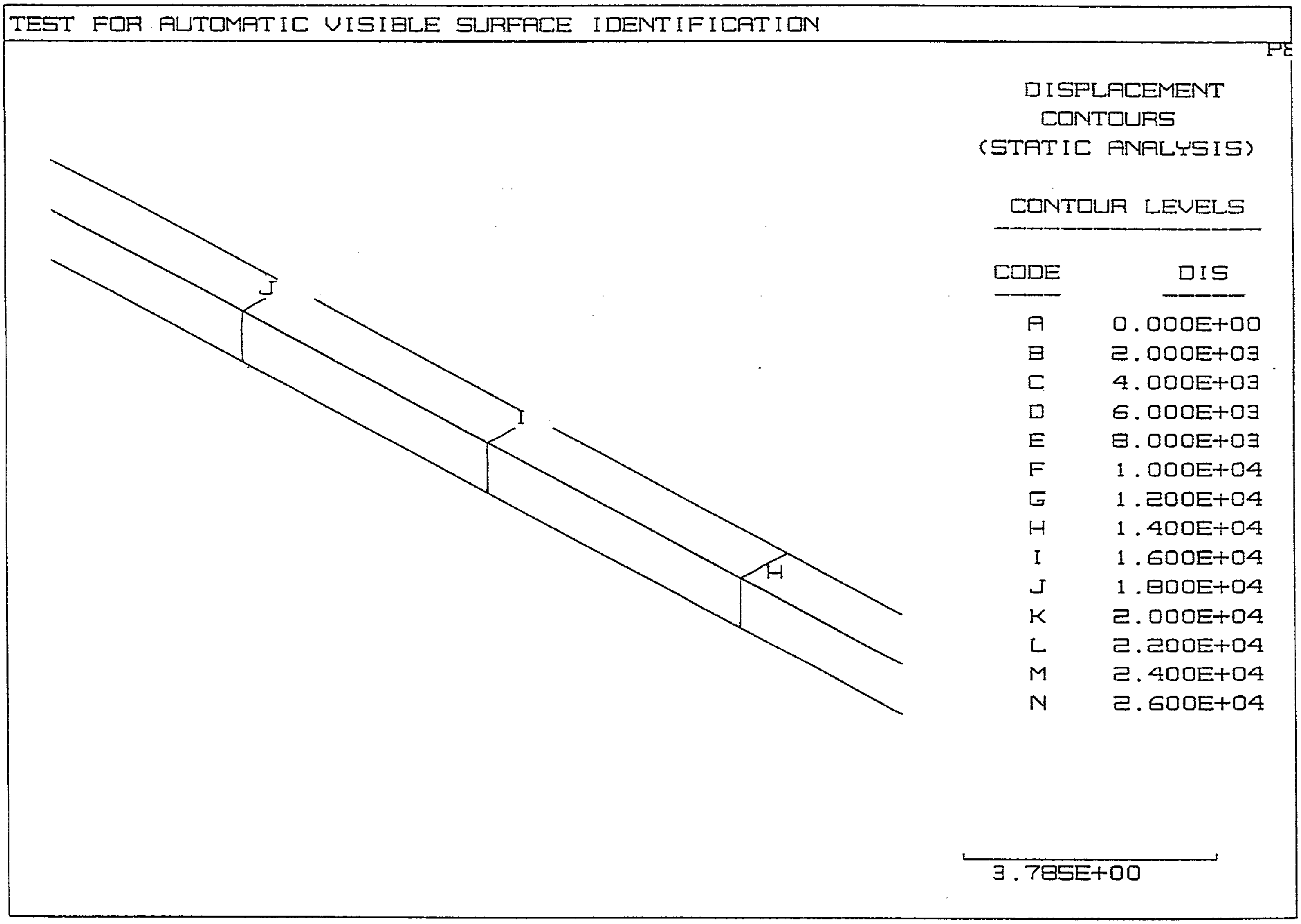


FIGURE 7.6: Windowed Section

CHAPTER 8

ACCOUNTING FOR GLOBAL NODAL ROTATIONS IN DEFORMED MODEL PLOTS FOR THE THICK-THIN SHELL ELEMENT

8.1 Introduction

The rotational degrees-of-freedom for the VAST thick-thin shell element (IEC=1) and the transition element (IEC=6) can be defined either with respect to the global (X,Y,Z) coordinate directions or with respect to the local $(\bar{V}_1, \bar{V}_2)$ coordinate directions. This flexibility was not fully accounted for in VASHID. More specifically, the AUXM1 subroutine of module VASSUB, which calculates the upper and lower coordinates from mid-surface coordinates, and upper and lower deflections from mid-surface deflections for the thick-thin shell element (IEC=1 modelling option 2) and the transition (IEC=6) previously accounted for only the rotations expressed in the local coordinate system. The purpose of work conducted under this contract was to correctly account for the nodal rotations defined by the user in the global XYZ coordinate system as well.

8.2 Theoretical Considerations

The theory which underlines the algorithm used in the AUXM1 subroutine for correctly accounting the nodal rotations expressed with respect to the global XYZ system is explained in the following section. The integer variable called IGROT is used to determine whether the rotational degrees-of-freedom are to be expressed in local or global coordinate systems. If the global XYZ coordinate system is desired, IGROT will be set equal to 1. The default value of IGROT is 0 which implies that the local coordinate system is chosen.

The displacement at any point within a shell element can be defined by the three Cartesian components of the mid-surface nodal displacements (u_i, v_i, w_i) and two rotations of the nodal vector $\bar{V}_{3i}$ about orthogonal directions ($\bar{V}_{1i}$ and $\bar{V}_{2i}$) normal to it, i.e.

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \Sigma\, N_i \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}_{mid} + \Sigma N_i \varsigma\, \frac{t_i}{2} \left[\bar{V}_{1i} \;\; -\bar{V}_{2i}\right] \begin{Bmatrix} \alpha_i \\ \beta_i \end{Bmatrix} \tag{8-1}$$

in which u,v, w = displacement in the xyz coordinate system; t_i = shell thickness at node i; $\bar{V}_{1i}, \bar{V}_{2i}$ = orthogonal unit vectors normal to $\bar{V}_{3i}$; and α_i, β_i = scalar rotations about vectors $\bar{V}_{2i}$ and $\bar{V}_{1i}$, respectively (see Figure 8-1).

The rotational degrees-of-freedom for shell element (IEC=1) and the transition element (IEC=6) defined in the global XYZ coordinate system (IGROT=1) is shown in Figure 8-2. It should be noted that $\bar{V}_3$ is the unit vector connecting the upper and lower surfaces and the mid-surface coordinates of shell elements; and $\theta_x, \theta_y, \theta_z$ are the rotational degrees of freedom at the midsurface node about the global X, Y, and Z axes, respectively.

The translations u_e, v_e, w_e, in the direction of X, Y, and Z axes due to the nodal rotations ($\theta_x, \theta_y, \theta_z$) can be calculated as follows:

$$u_e = \frac{t_i}{2} \varsigma \, [V_3(3)\, \theta_y - V_3(2)\, \theta_z] \tag{8-2}$$

$$v_e = \frac{t_i}{2} \varsigma \, [-V_3(3)\, \theta_x + V_3(1)\, \theta_z] \tag{8-3}$$

$$w_e = \frac{t_i}{2} \varsigma \, [-V_3(2)\, \theta_x - V_3(1)\, \theta_y] \tag{8-4}$$

where ς is +1 for upper surface nodes and -1 for lower surface nodes. From vector analysis, the translation (u_e, v_e, w_e) due to nodal rotations ($\theta_x, \theta_y, \theta_z$) defined with respect to the global XYZ coordinate system can be expressed in terms of the cross-product of $\bar{\theta}$ and $\bar{V}_3$ as follows:

$$\bar{u} = \frac{t_i}{2} \varsigma \, [\bar{\theta} \times \bar{V}_3] \tag{8-5}$$

where, in vector form:

$$\bar{\theta} = \theta_x \bar{i} + \theta_y \bar{j} + \theta_z \bar{k} \tag{8-6}$$

$$\bar{V}_3 = V_3(1)\bar{i} + V_3(2)\bar{j} + V_3(3)\bar{k} \tag{8-7}$$

and

$$\bar{u} = \frac{t_i}{2}\zeta \, [u_e\bar{i} + v_e\bar{j} + w_e\bar{k}] \tag{8-8}$$

Now the displacement at any point within the shell element can be defined by the three Cartesian components of the mid surface nodal displacements and three global rotations $\theta_x, \theta_y, \theta_z$, about the global X, Y, and Z axes, i.e.

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \Sigma \, N_i \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}_{mid} + \Sigma \, \frac{t_i}{2} \, \zeta \begin{Bmatrix} u_e \\ v_e \\ w_e \end{Bmatrix} \tag{8-9}$$

8.3 Implementational Details

The implementation of the theory described in the previous section in the AUXM1 subroutine of module VASSUB has required that the integer variable, IGROT, to be read from the PREFX.GOM file and transferred to the NTS1 scratch file. This has been done in the SGEOM subroutine of module VASSUB. The element data, and the value of IGROT is read by the VASHID program from the NTS1 scratch file and the value of IGROT is then transferred to the AUXM1 subroutine through its argument list.

To validate the implementation of the global rotation option, a centilevered beam problem was modelled with the thick/thin shell elements as shown in Figure 8.3. A transverse end load was applied. Deformed model plots produced by PLOTV2, MOVIE, VASHX and VASHP are shown in Figures 8.4 to 8.7, respectively. When the global rotation option is used with the shell element. For comparison purposes, the same problem was solved with the local option used with the shell element and the deformed model plots are shown in Figures 8.8 to 8.11.

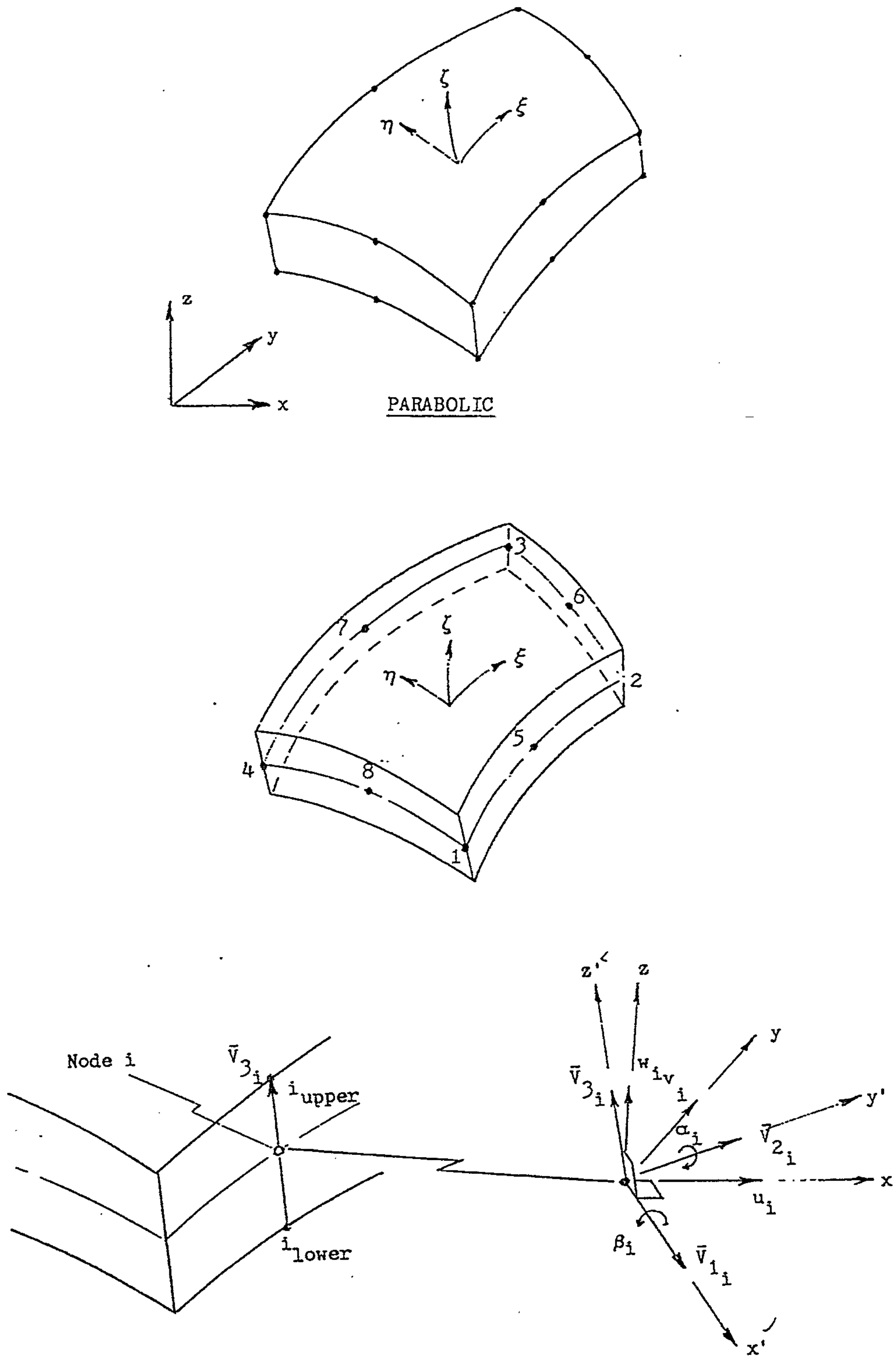


FIGURE 8.1: Curved Shell Elements

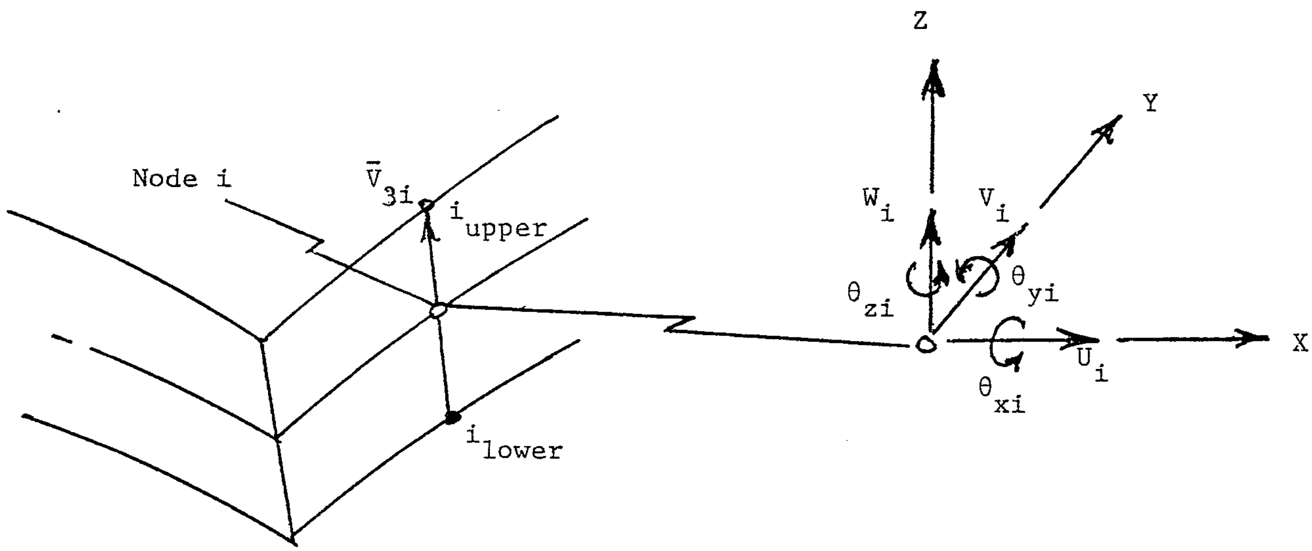


FIGURE 8.2: Sign Conventions for Global Rotation

8.6

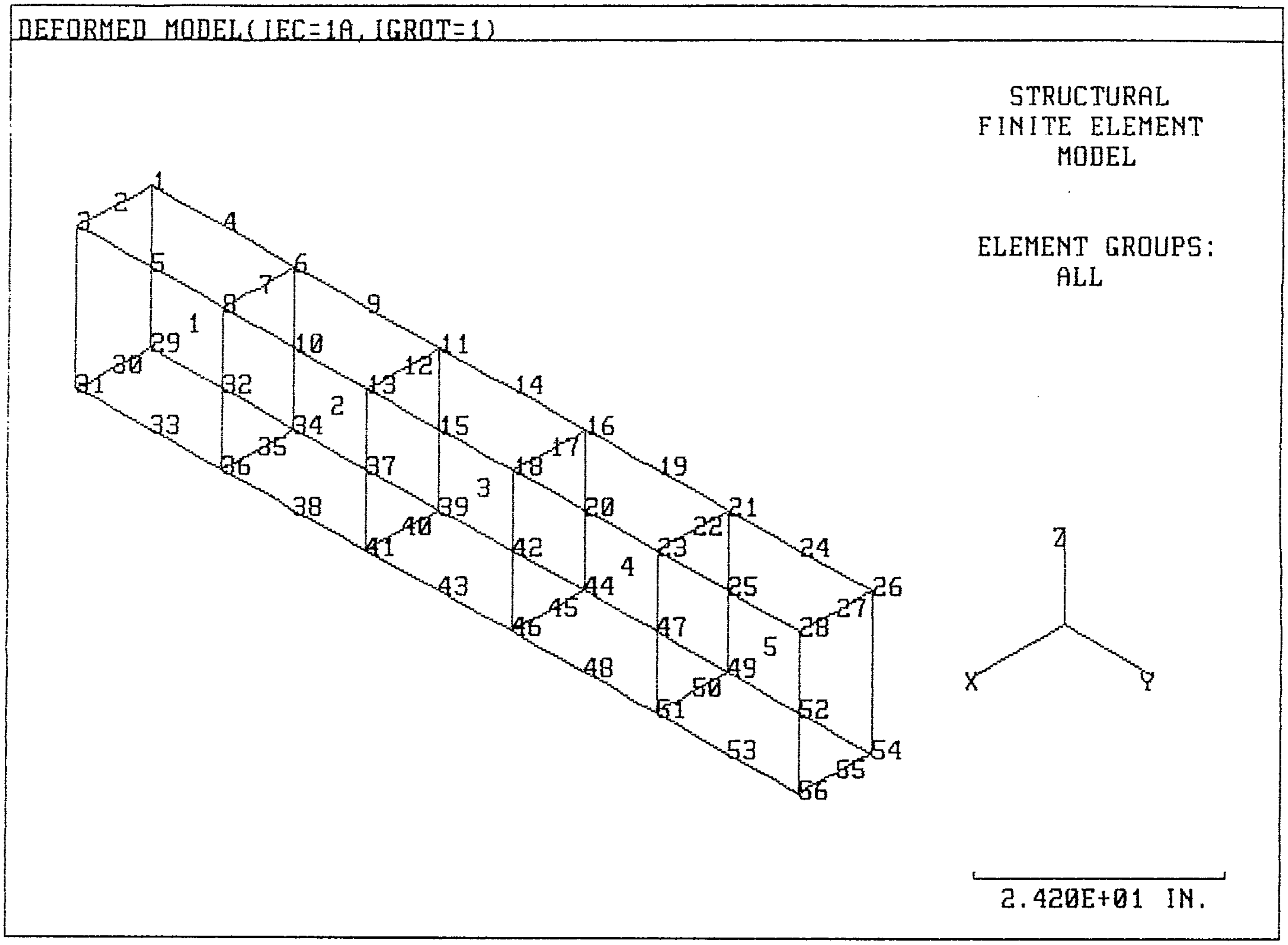


FIGURE 8.3: Centilevered Beam Problem Discretized using Thick/Thin Shell Elements

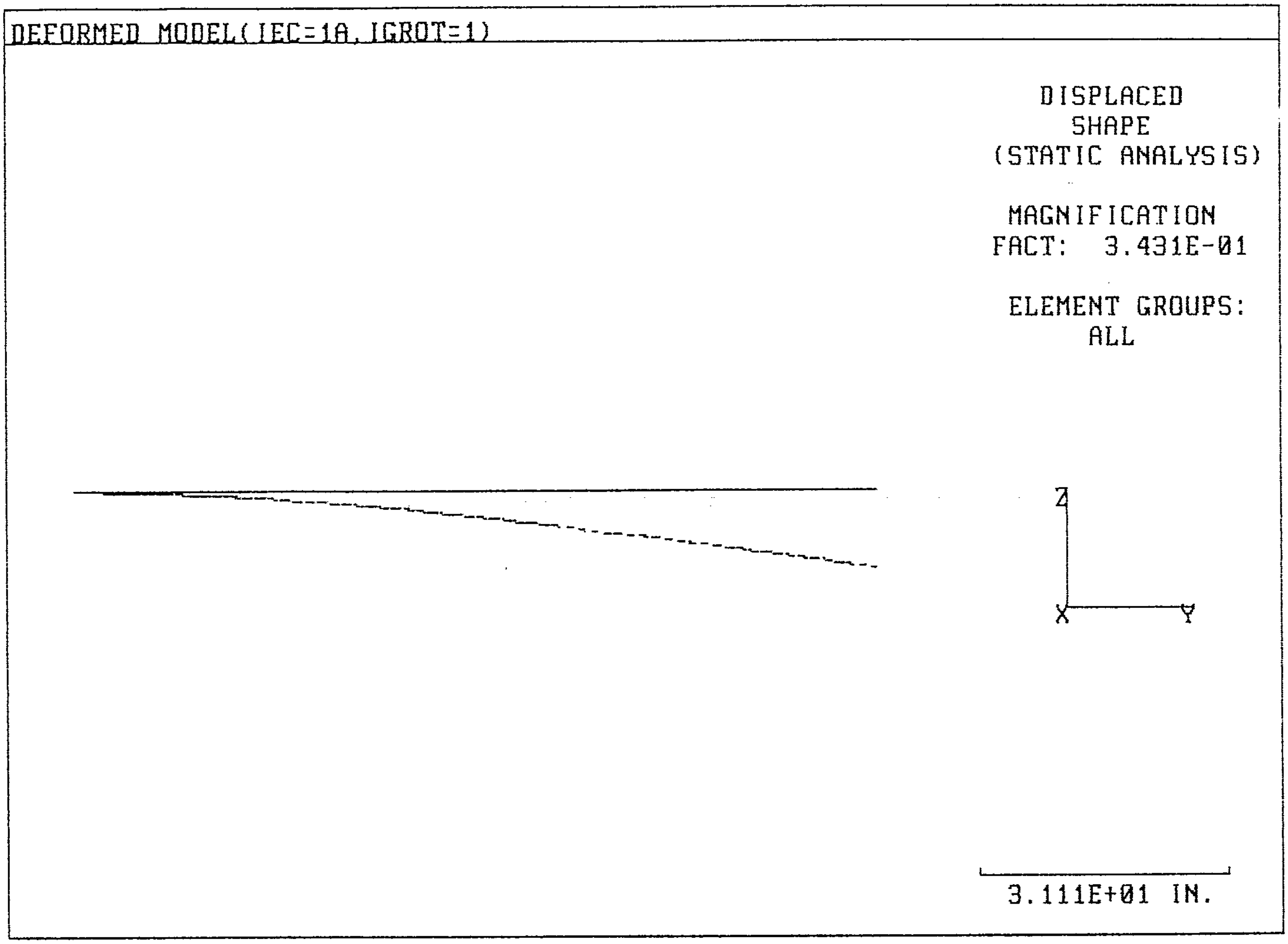


FIGURE 8.4: Deformed Model Plot Produced by PLOTV2 for Global Rotation Option (IGROT=1)

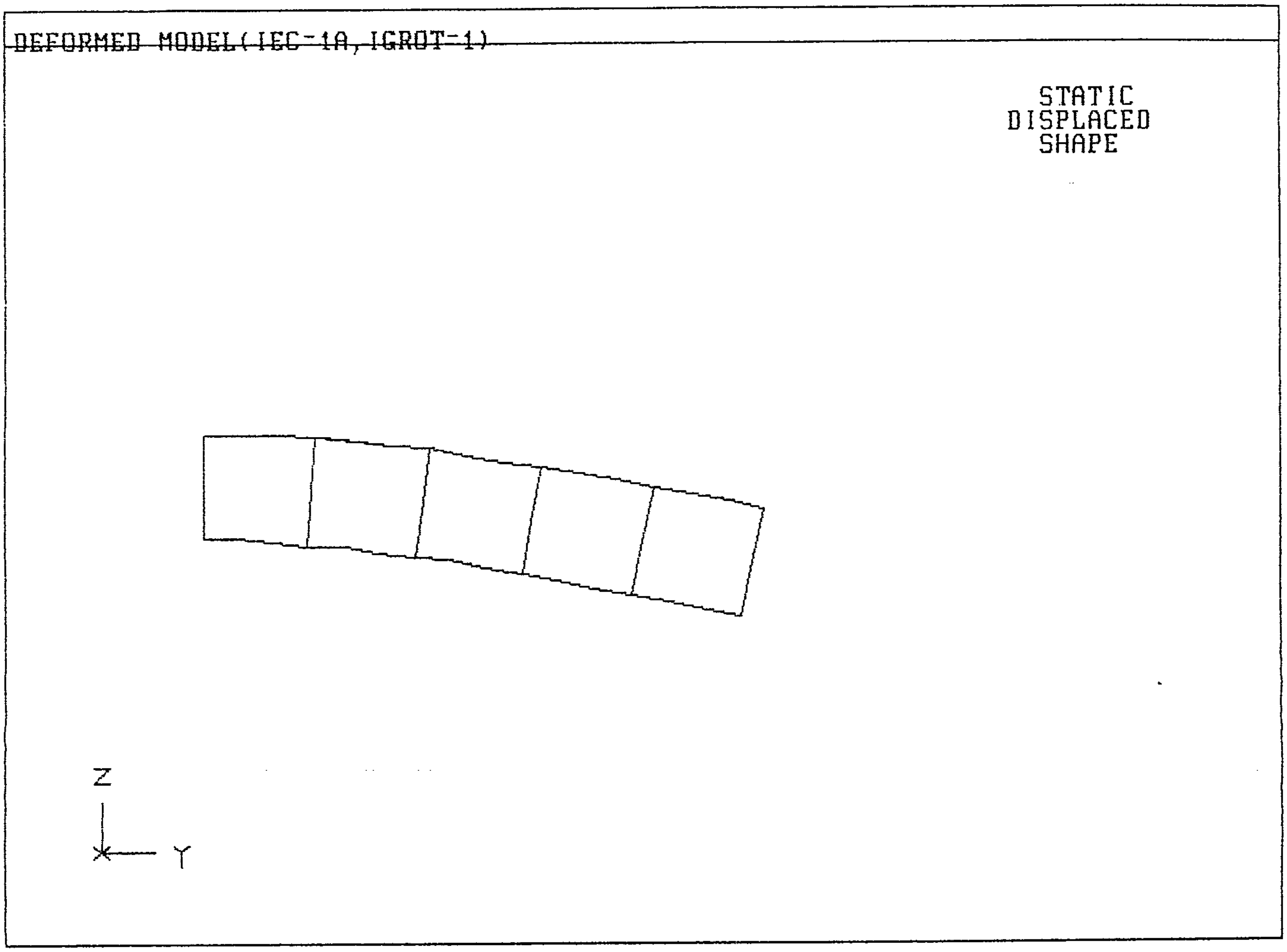


FIGURE 8.5: Deformed Model Plot Produced by MOVIE for Global Rotation Option (IGROT=1)

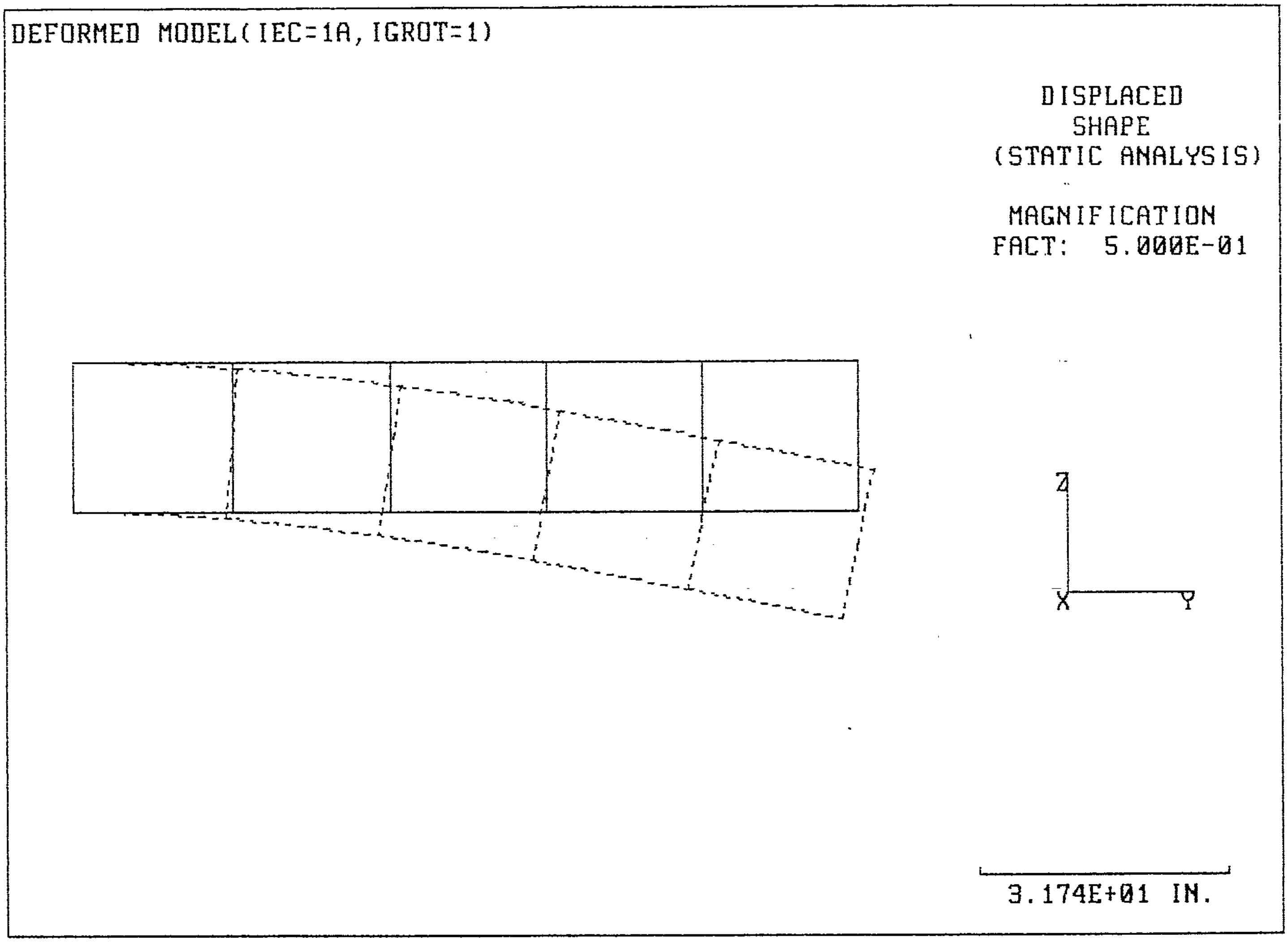


FIGURE 8.6: Deformed Model Plot Produced by VASHX for Global Rotation Option (IGROT=1)

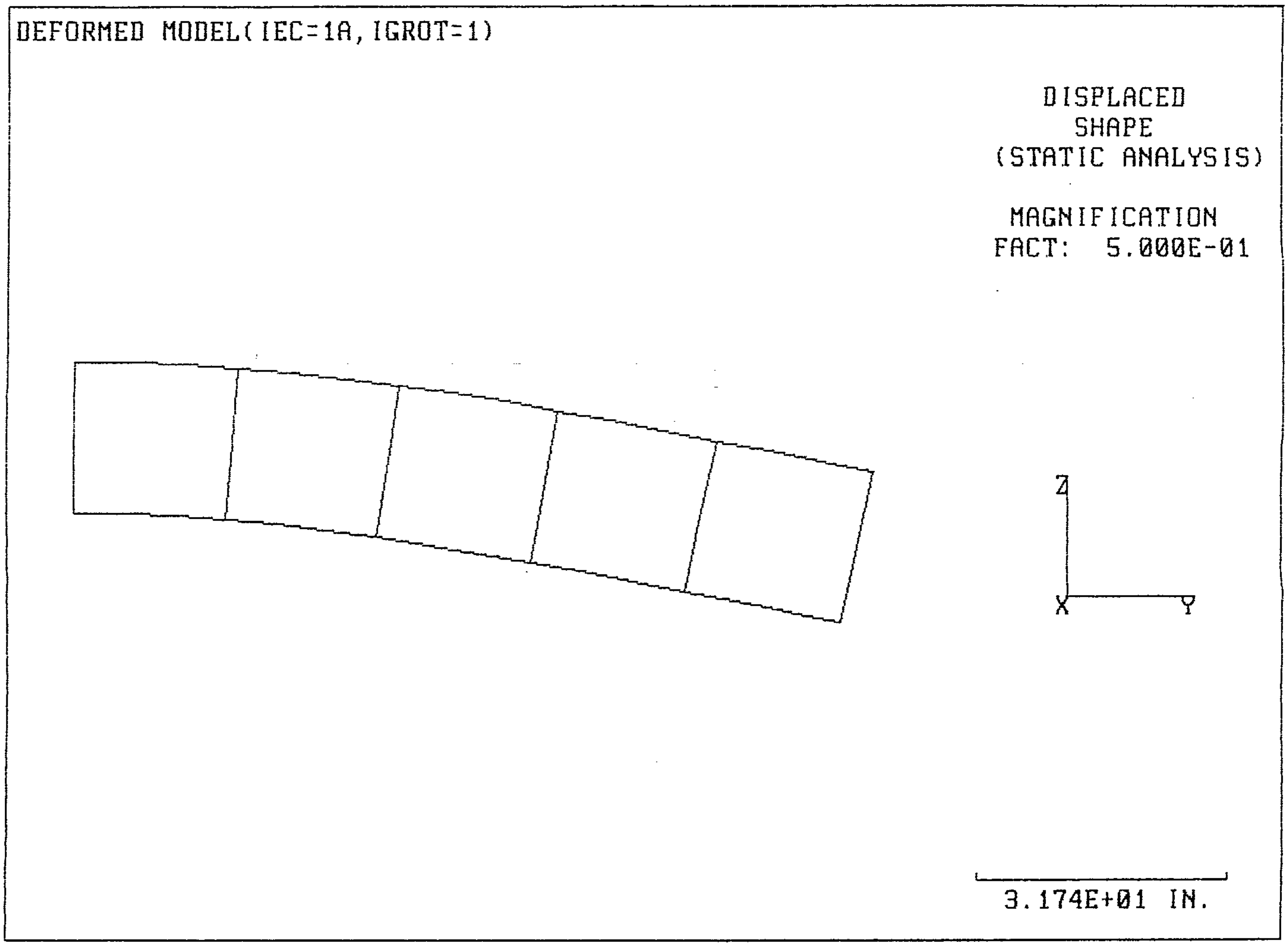


FIGURE 8.7: Deformed Model Plot Produced by VASHP for Global Rotation Option (IGROT=1)

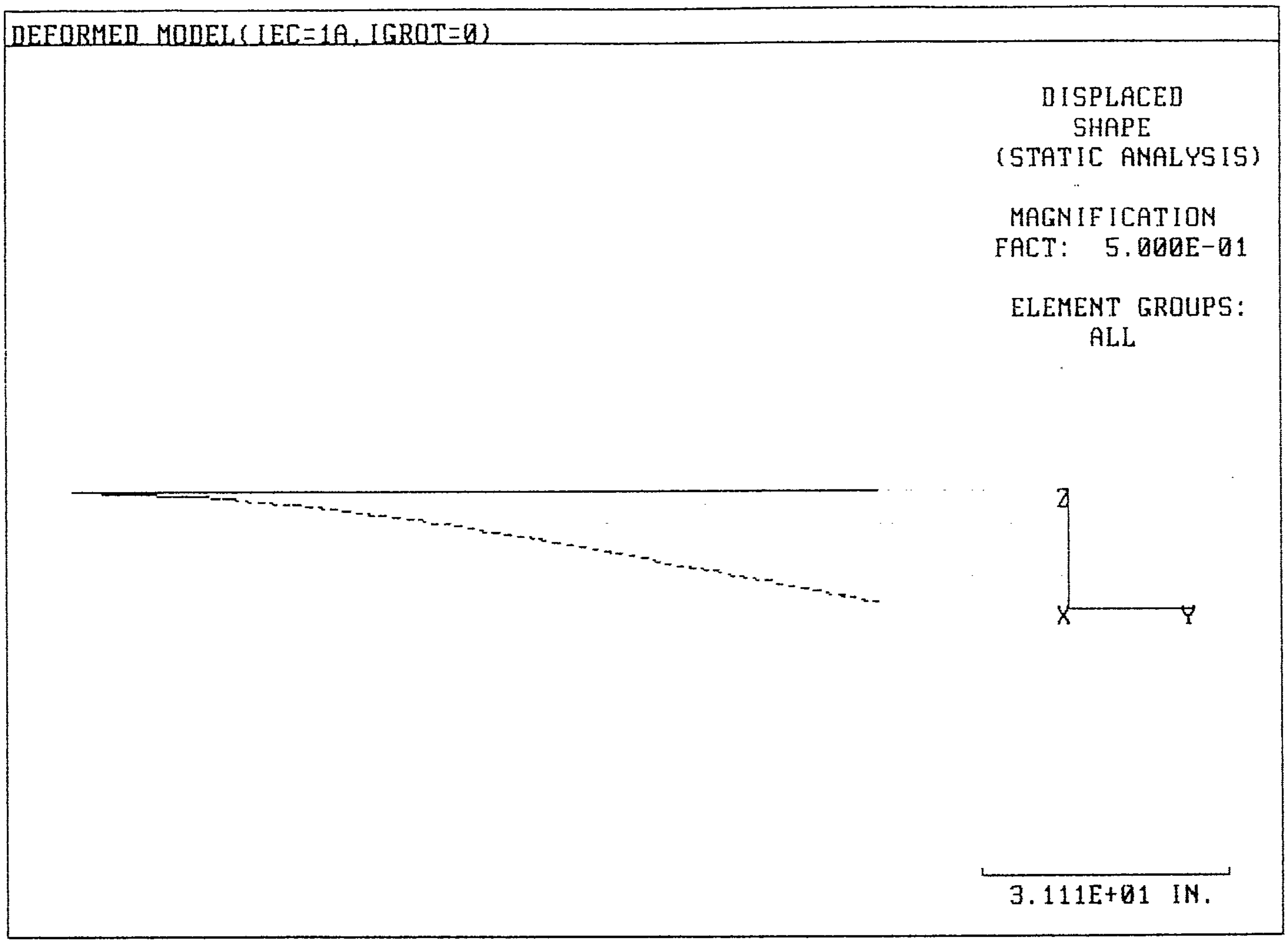


FIGURE 8.8: Deformed Model Plot Produced by PLOTV2 for Local Rotation Option (IGROT=0)

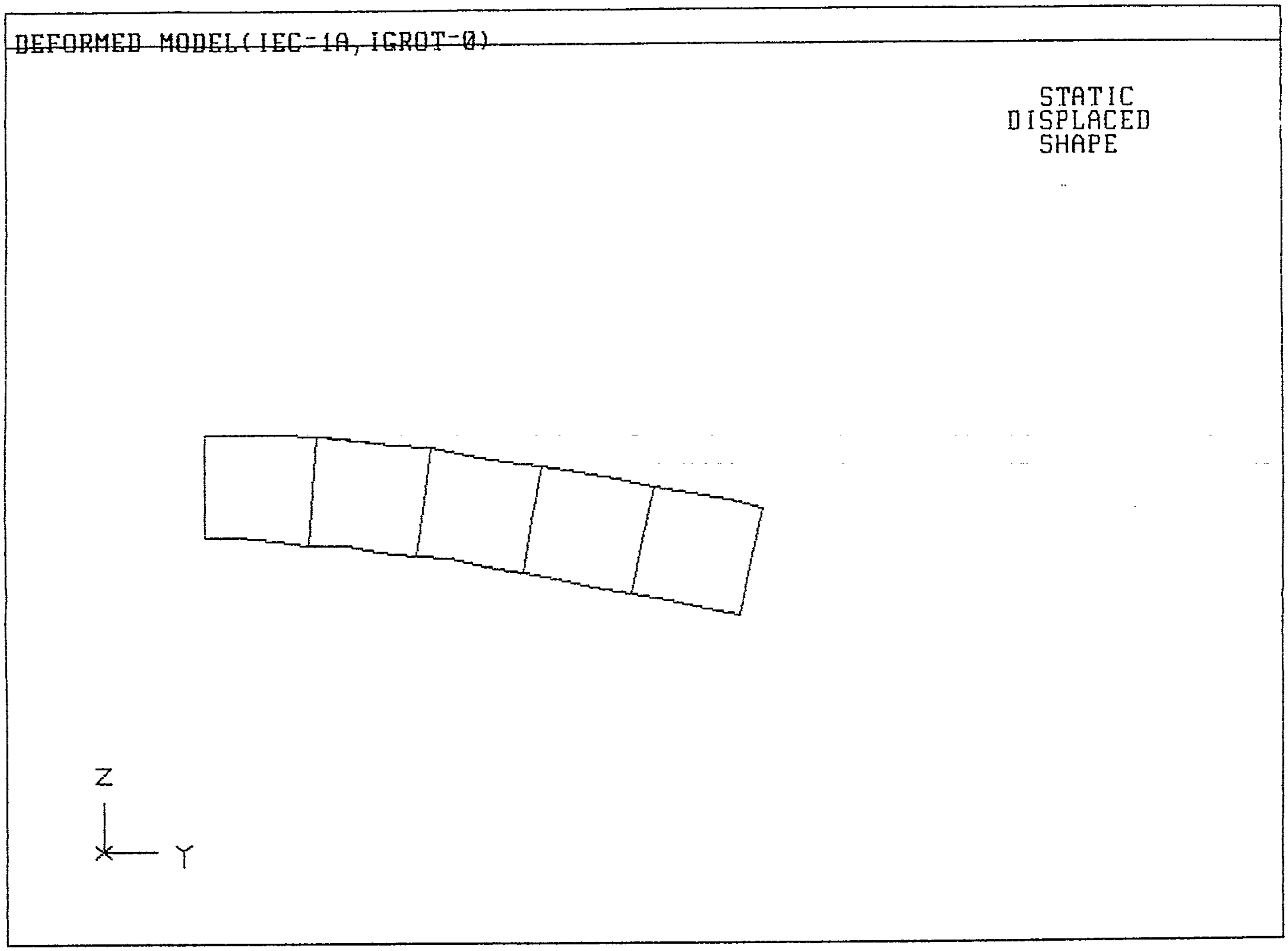


FIGURE 8.9: Deformed Model Plot Produced by MOVIE for Local Rotation Option (IGROT=0)

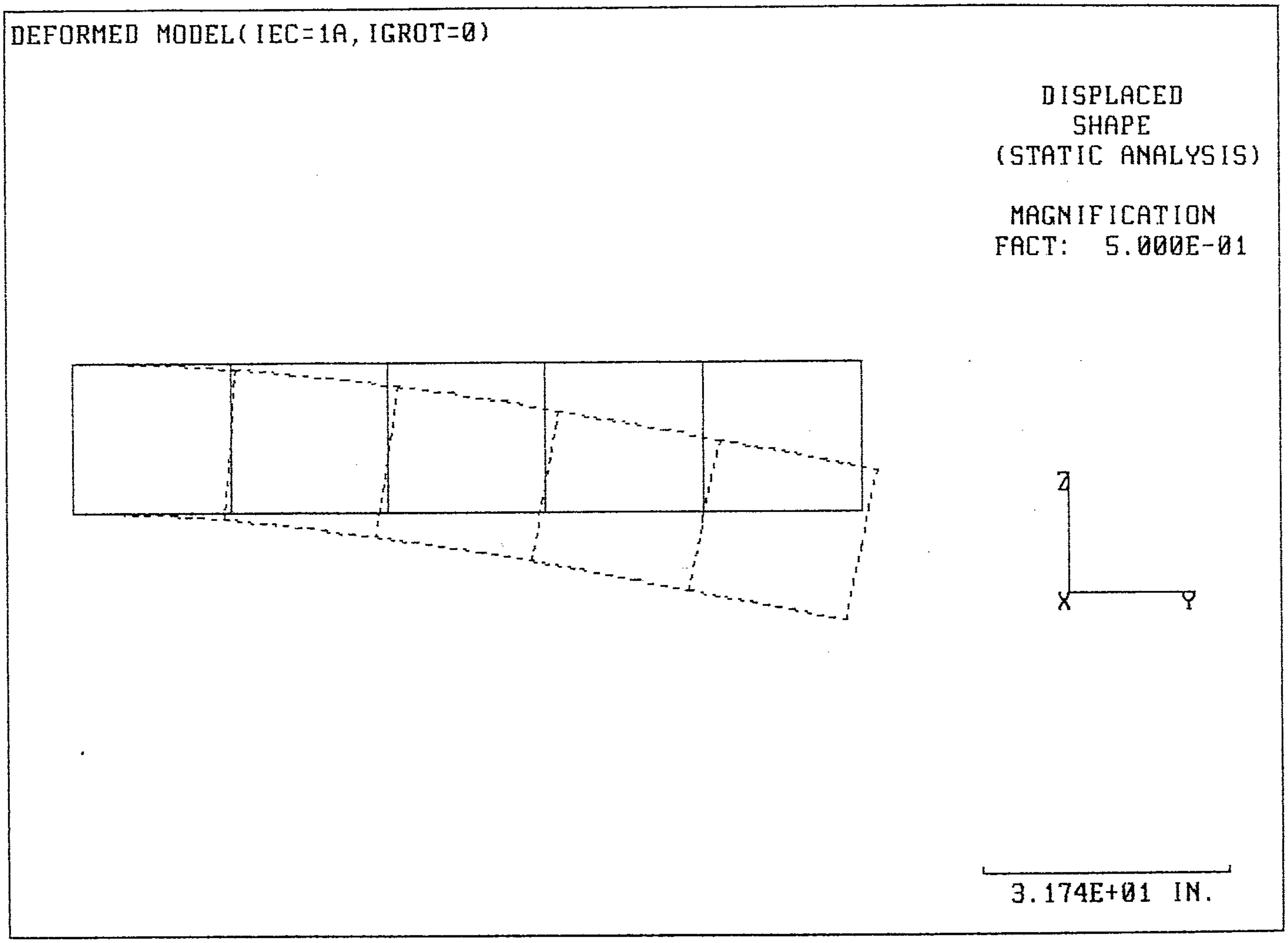


FIGURE 8.10: Deformed Model Plot Produced by VASHX for Local Rotation Option (IGROT=0)

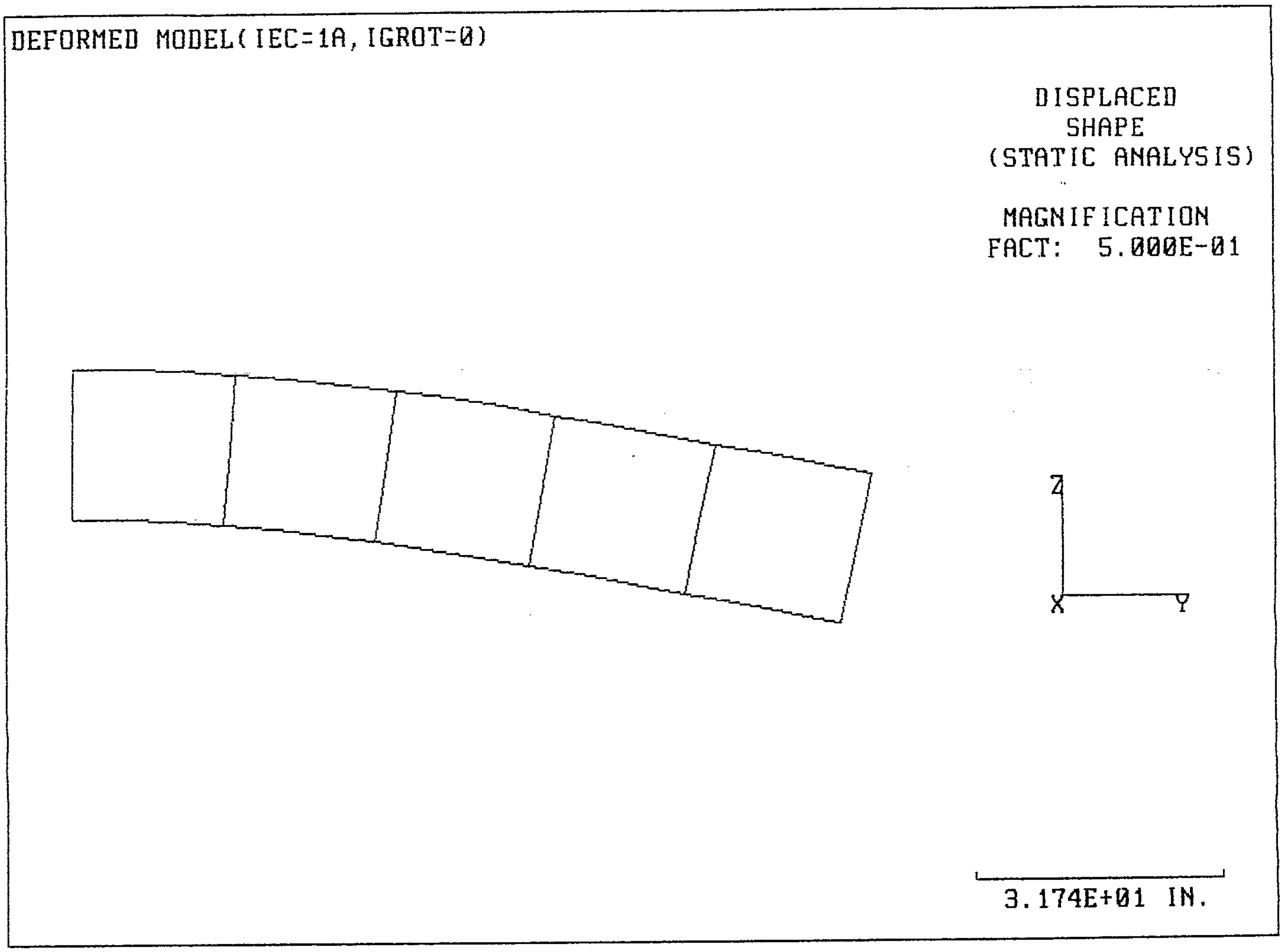


FIGURE 8.11: Deformed Model Plot Produced by VASHP for Local Rotation Option (IGROT=0)

CHAPTER 9

IMPROVED USER CONTROL ON THE BEAM ELEMENTS TO BE PLOTTED BY PLTV16

The PLTV16 program, which displays beam stresses, has been upgraded to the VASTG standard so that the user will control the elements to be displayed. The previous version of the program plotted one superelement at a time and all the beam elements were displayed.

This required improvement in control over elements to be displayed involved modification of the PLTV16 program to call the PLOTV library subroutine ELOPT. This subroutine contains the standard prompting for the element group and range of elements to be plotted or to specify the element types to be plotted. The data is placed into the array IOPT2 and the PLTV16 program was modified to display the stresses, bending moments, axial and shear forces of the beam elements as identified in this array.

An illustration of the improvement made in the PLTV16 can be seen in the following example of a plane truss which is made up of 5 element groups. Figure 9-1 shows the bending moments of beam elements when all the element groups are selected. Figure 9-2 shows the bending moments when all of the elements only in the element group #1 are selected.

9.2

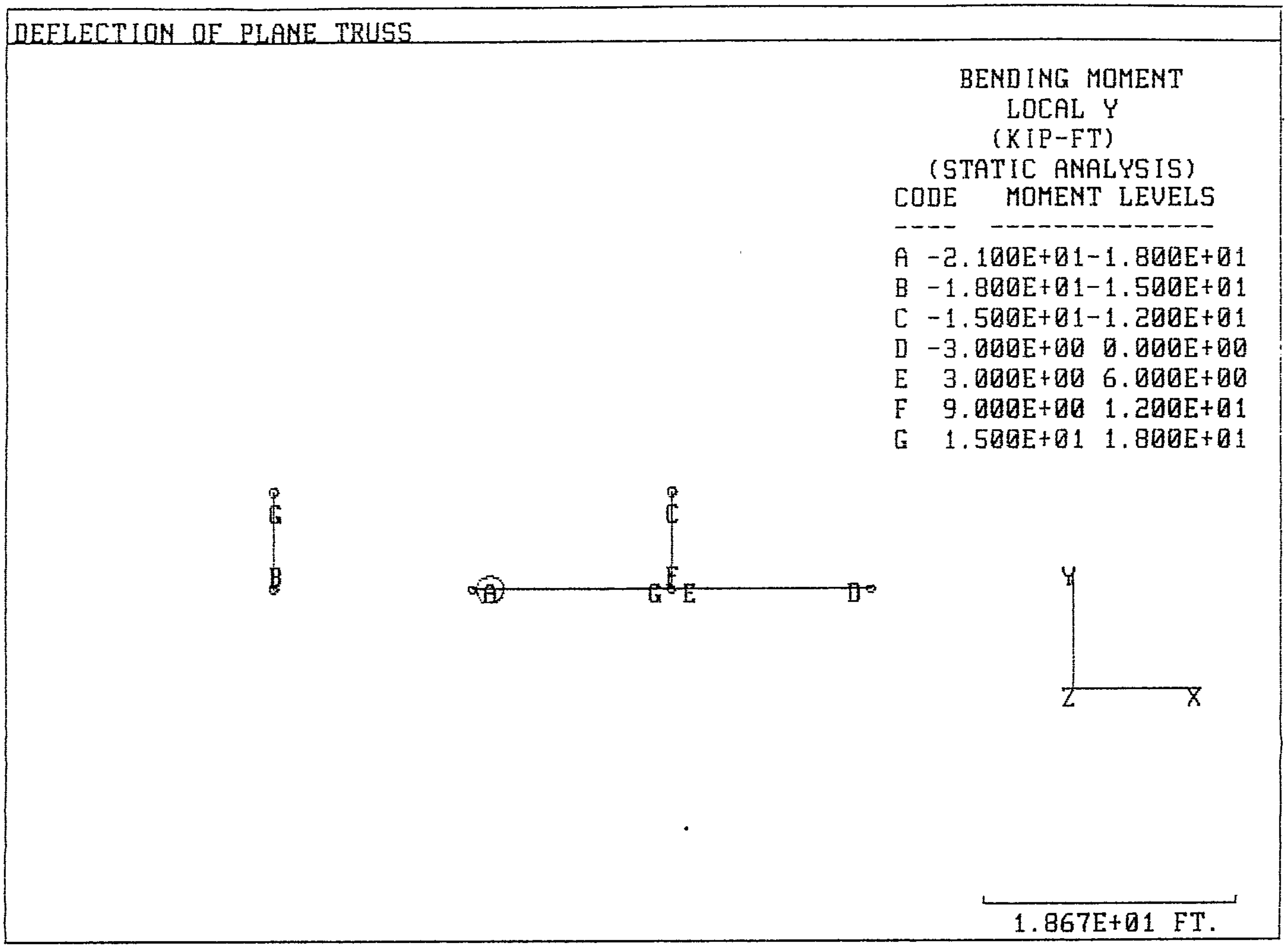


FIGURE 9.1: Bending Moments of Beam Elements
in all Five Element Groups

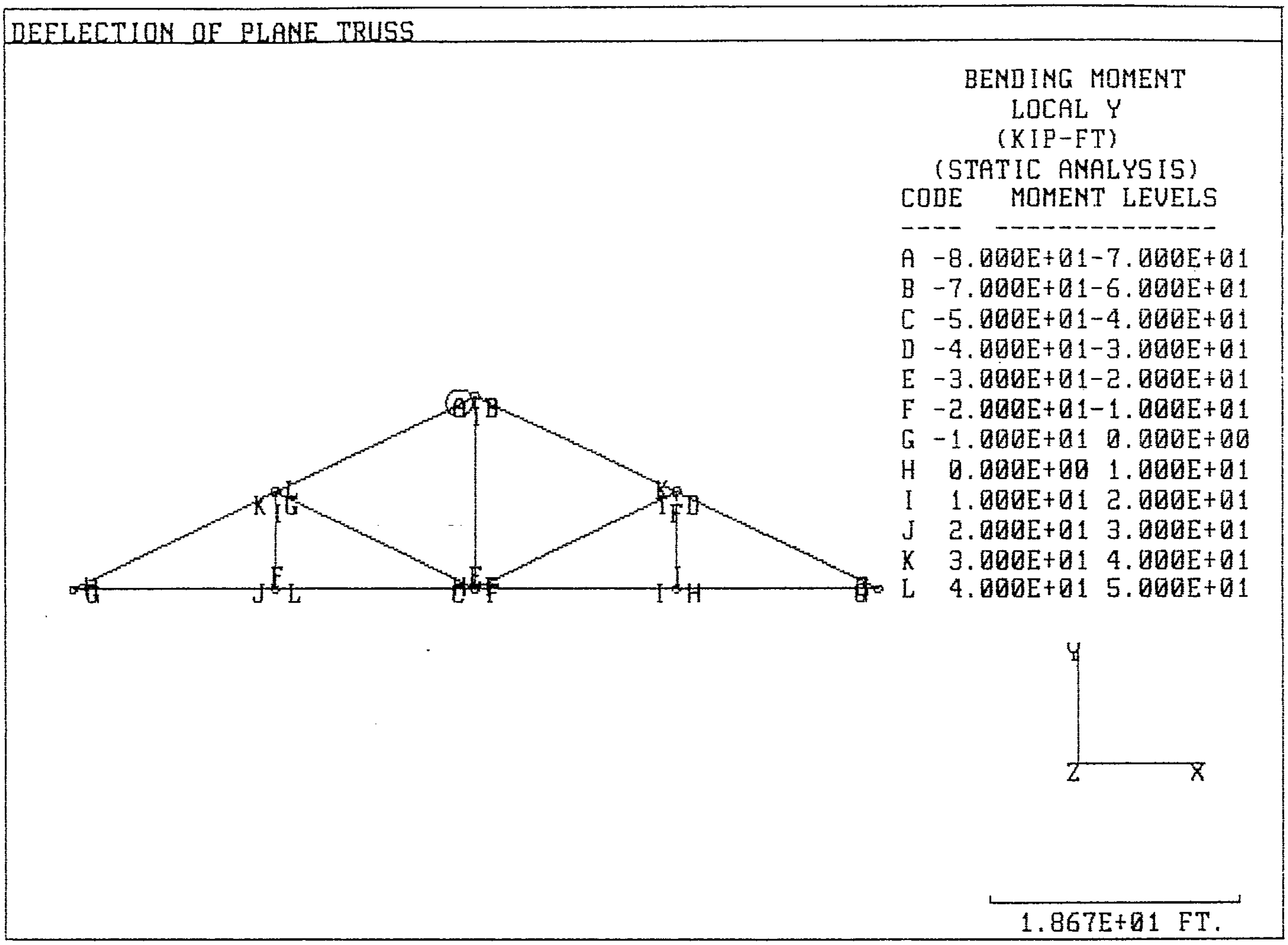


FIGURE 9.2: Bending Moments of Beam Elements
in Element Group 1

CHAPTER 10

WINDOW DIMENSIONS CONTROLLED BY PLOTTING SPECIFICATIONS

10.1 Introduction

Scaling parameters for geometry plots have been determined in VASTG from the coordinates of all model nodes whether they were associated with the part to be plotted or not. To "blow-up" or expand the plot necessitated the use of the graphics cursor option 'WINDOW'. It should be noted that the 'WINDOW' feature is available only on terminals with cursor capabilities and it can be activated only once the plotting of the unexpanded plot is complete. This represented a major inconvenience to the analyst who in most occasions would prefer to have the plots scaled to fit the screen based on element selections made.

The implementation within VASTG of the capability to automatically scale a plot to fit the screen when only part of the model is selected for viewing is described in the following section. As might be expected the new feature implemented under this contract results in a savings of time since the need the plotting of the extra unexpanded plot is removed as are the steps involved in the window option. This feature also proves to be very useful when producing printer hardcopies on the PC version of the program. The generation of this printer hardcopy file results in the plot not being displayed on the monitor so that the capability to window is not available. An example is also given to illustrate the use of the new capability.

10.2 Implementation

Recall that the IOPT2 array is defined by the PLOTV library subroutine ELOPT which prompts for the element plotting specifications in a standard form. The format of the IOPT2 array is given in Table 302.1. A general purpose subroutine, SPLOT, to determine the user window dimensions for plotting based upon the user supplied element plotting specifications contained in the IOPT2 array has been developed. The VASTG model plotting modules utilizing the ELOPT subroutine have also been provided with the capability to use the SPLOT routine.

10.2

The seven VASTG modules that have been modified to use the new subroutine SPLOT are identified in Table 10.1. Implementation of this feature within the seven VASTG module required new prompting and the calls to SPLOT. Modules with the capability to plot with default specifications had additional changes in their subroutines for generating the session file containing the default specifications. The default specification will be that the window dimensions will not be controlled by plotting specifications. For modules using the command directives, a new directive 'SCALE' was added to activate/deactivate this capability.

10.3 Example

The plot shown in Figure 10.1 is that of an unsubstructured model containing 18 shell elements. The plot was generated by using the pre-set defaults and entering the command directive 'PLOT'.

Figure 10.2 is a plot of 2 elements only. It was generated by entering the command directive 'ELEM'. The resulting prompts and responses to display only elements 4 and 5 of the model are listed below.

```
DO YOU WANT A SUMMARY OF ELEMENTS AND GROUPS? (O=NO)
>O
DO YOU WANT ALL THE ELEMENTS PLOTTED? (O=NO)
>O
ENTER   0 TO SPECIFY ELEMENTS BY GROUP AND RANGE
        1 TO SPECIFY ELEMENTS BY TYPE.
>O
SPECIFY ELEMENT GROUP AND FIRST AND LAST ELEMENTS TO BE PLOTTED
(ENTER O,O,O TO TERMINATE).
>1,4,5
>O,O,O
```

Figure 10.3 is a plot with scale based on the portion of the model selected for viewing. This plot was generated by entering the 'SCAL' directive and responding '1=YES' to the prompt.

TABLE 10.1: PLOTV Modules Having Scaling Capability

PLOTTING PROGRAM	FUNCTION
PLOTV1	to plot structural F.E. model
PLOTV2	to plot deflected shapes
PLOTV3	to plot stress/strain contours
PLOTV4	to plot principal stress or strain vectors
PLOTV9	to plot eigenmode of displacement contours
PLOTV11	to plot strain energy density contours
PLOTV12	to plot applied structural loads

10.4

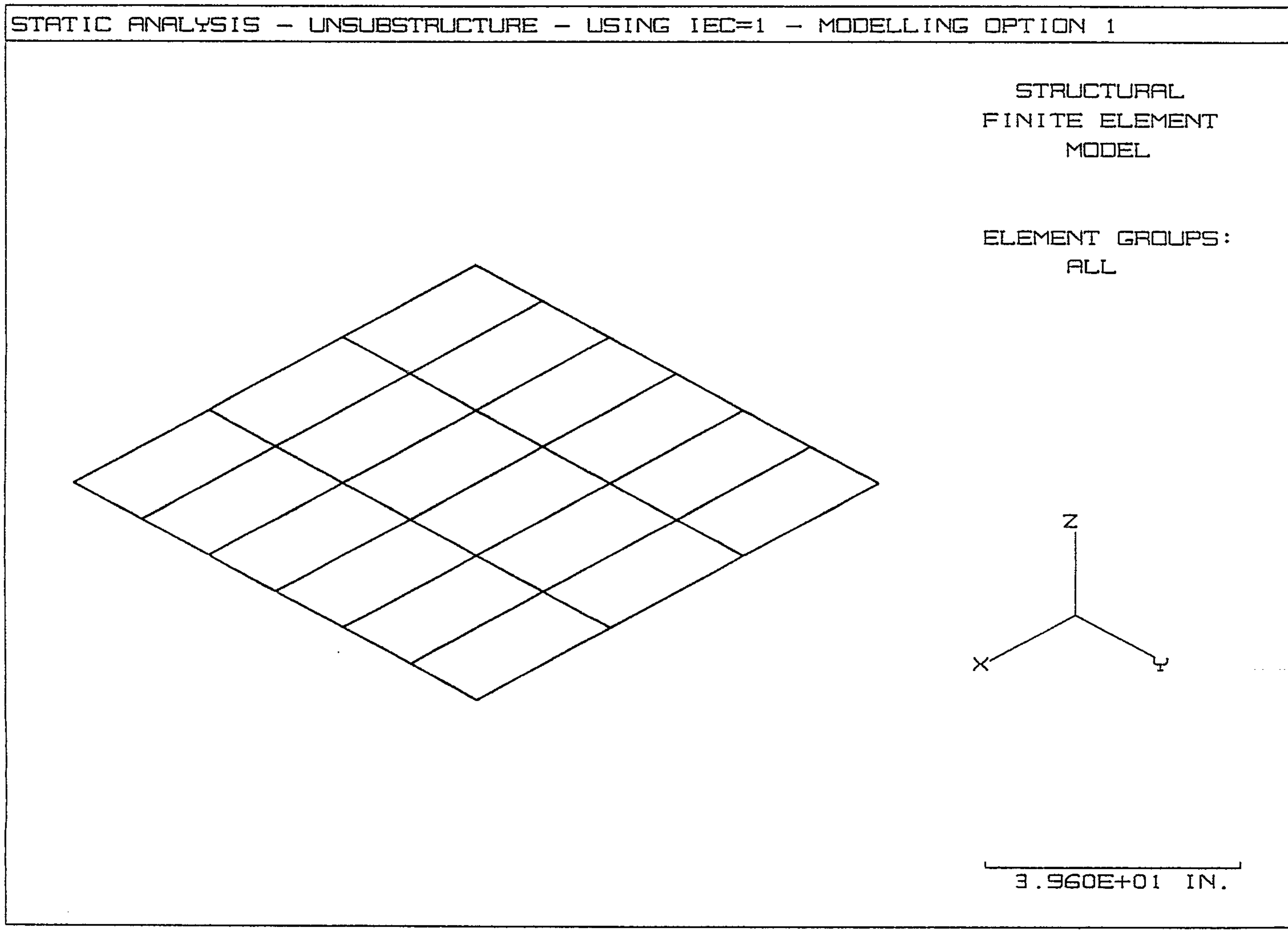


FIGURE 10.1: Plot of Full Unsubstructured Shell Finite Element Model

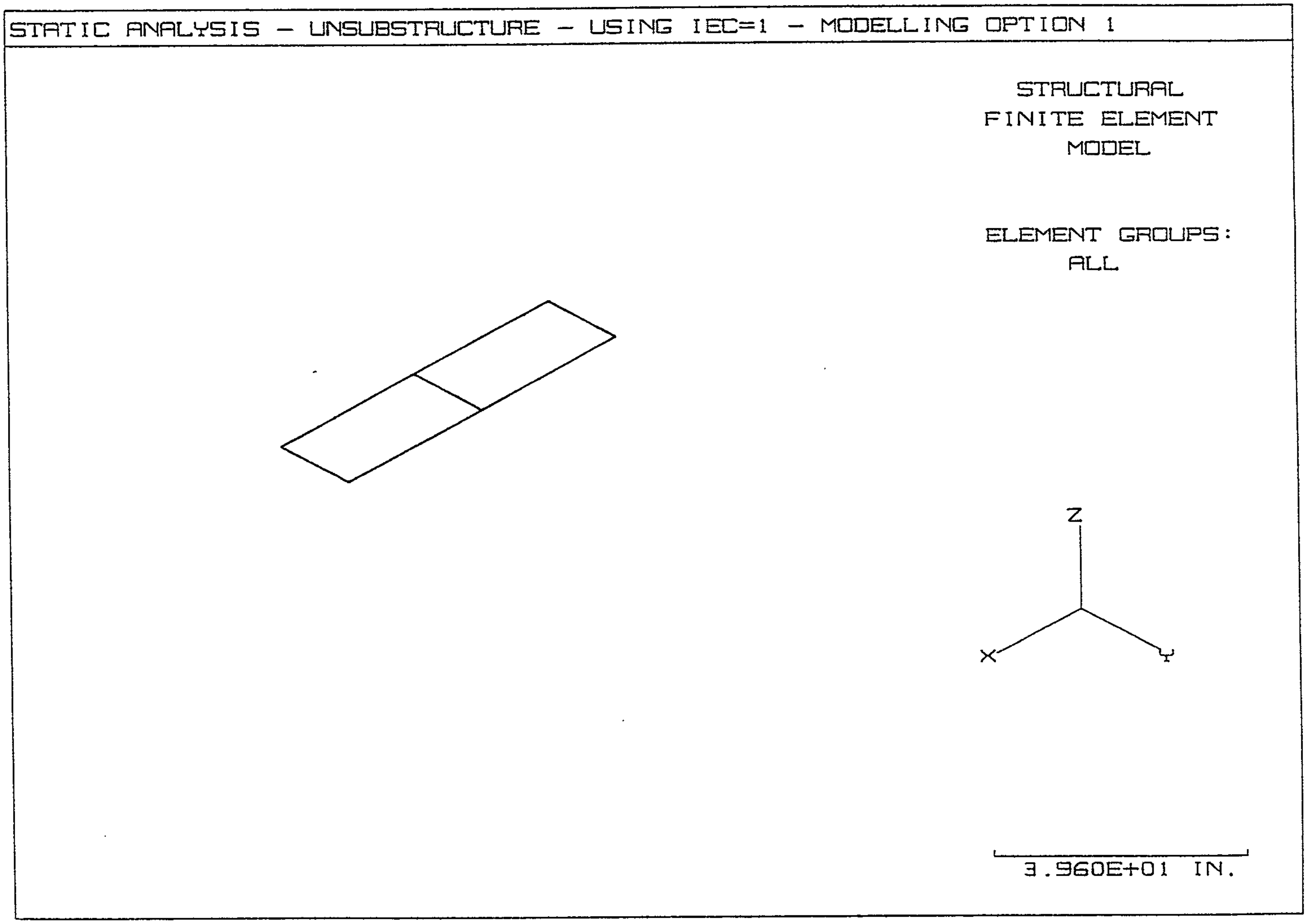


FIGURE 10.2: Plot of Two Elements of the Unsubstructured Shell Finite Element Model

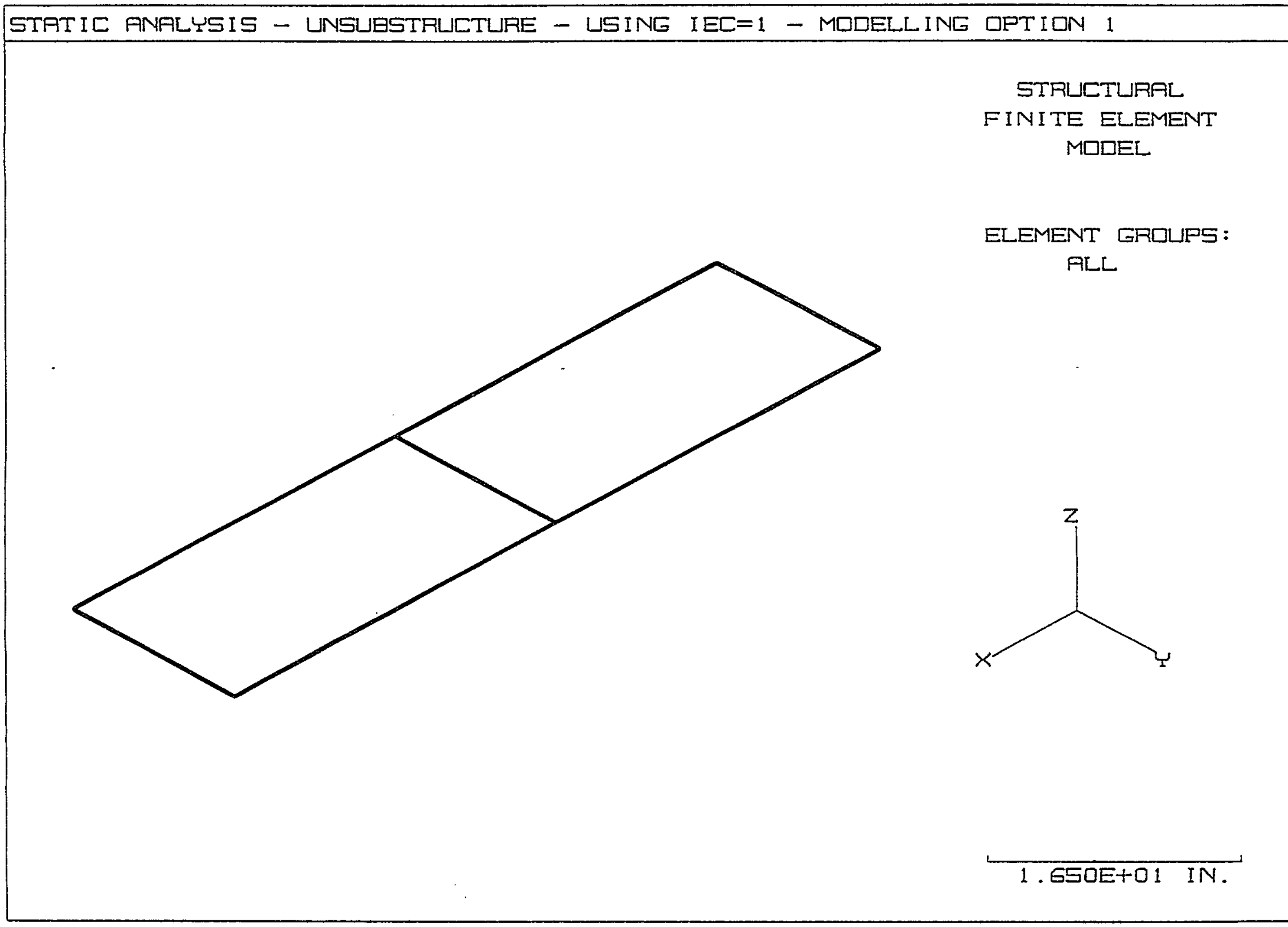


FIGURE 10.3: Plot of Two Elements of the Unsubstructured Shell Finite Element Model Using the Option to Scale to Full Screen Size

CHAPTER 11
MODIFICATIONS TO THE PLOTVX LIBRARY

11.1 Introduction

The PLOTVX library contains the generic subroutines which perform the plotting, terminal control functions, and any function specific to the operating system. All hardware and graphics language dependence is contained within this library.

Under this contract, the capability to save a laser plot file has been investigated and implemented. This capability will be discussed in Section 11.2 and is available on the mainframe versions of VASTG. This feature was implemented through the PLOTVX library subroutine, TERCTN, which controls the terminal functions.

The PLOTVX library subroutines FACLTY and TERCTN were reviewed and restructured under this contract and the work performed is described in Section 11.3.

11.2 Saving a Laser PLOT File

The laser printer produces a high quality hardcopy plot as compared to a plot produced by a Tektronix pen plotter or a screen dump to a dot matrix printer. The implementation of the capability to save a laser plot file makes it easy for users to obtain report quality plots. The previous approach for generating a laser plot involved the capture of the PLOT10 generated output stream that leaves the terminal to a file, the editing of the file to remove extraneous data, and the submission of the file to the laser printer for hardcopy printing. Such a technique was therefore only possible by an experienced PLOT10 user proficient with VAX/VMS editors. The new technique requires only the entry of 'S' to save the plot following its display and the correctly formatted file for a laser plotter is generated.

11.2.1 Implementation

The modifications necessary to implement capability to generate a laser print plot file were as follows:

1. Development of a new entry point for the PLOTVX subroutine TERCTN. This modification was performed in both the PLOT10 main frame and the GKS-PC versions of the PLOTVX module. TERCTN is the routine that controls how the terminal functions. This routine has two arguments. The first, called the entry argument, identifies the function to be performed and the second argument may be either input or output depending on the function.

   The new entry point of 7 performs various operations on the printer plot file, depending upon the value of the subroutine argument IARG. The printer plot file is rewound, opened, closed (and saved), or closed (and deleted), when IARG is equal to -1, 0, 1, 2, respectively.

   The name of the printer plot files are PREFX.LXX where XX refers to the number of the plot saved during a VASTG execution. The plots are numbered sequentially from zero and the current plot number is saved in a new common block named PLTFIL. The plot number is displayed to the user prior to saving a plot.

2. Modifications to the individual plotting programs to:

   (a) Set ISEG parameter to 0 to indicate that the plotting file is to be opened.

   (b) Provide a new prompt S - TO SAVE PLOT IN FILE, just prior to plot appearing on the screen.

   (c) Call to TERCTN with entry of 7 so as to open plotting file if ISEG=0 or to rewind the file if ISEG=-1.

   (d) Additional check if input is 'S' after the hold for input just prior to plot commencing. If the input is not 'S' the parameter ISEG is set to -1 for rewind when generating the next plot.

   (e) Call to TERCTN with entry of 7 to save file if the option 'S' is selected and then set ISEG=0 to open new plotting file.

   (f) Call to TERCTN with entry of 7 and ISEG=2 prior to exiting graphics mode value to indicate that the file is to be closed and deleted.

   The overall algorithm for laser plot saving is shown in Figure 11.1.

3. Modifications of the PLOT10 library internal subroutine TTOUT to duplicate synchronous I/O channel output to a disk file for subsequent spooling to a laser printer. Also modifications were required to save this file on unit 99 (arbitrary selected).

   This has identified a potential problem in that the PLOT10 source code utilized in creating the object file must be recreated. Some of the PLOT10 IGL subroutines cannot be found and efforts to locate them are still ongoing with MARTEC and DREA staff. As an interim measure, the PLOT10 library file was created by replacing the modified TTOUT subroutine in the object library which was created in May 1983 (the version linked into the VASTG6).

11.2.2 Program Operation

For main frame versions of VASTG, the synchronous device driver which drives the Tektronix terminal has been modified to create a disk file containing all the PLOT10 escape sequences. Following a plot the entry of 'S' <CR>; saves the plot file under the name PREFX.LXX where XX is 00 initially and is incremented by 1 for each 'S'ave command received during the execution of VASTG.

The file PREFX.LXX can be spooled to a printer which emulates a PLOT10 terminal/plotter. The user may use a system global symbol definition to allow a hard copy via PRINT_TEK.P. The generic command is:

PRINT/PARAM=(DATA_TYPE=TEK4014) PREFX.L00.

At the present time, colour escape sequences will create problems on the QMS Laser printer PLOT10 emulation firmware. Therefore, if this feature is to be used, the terminal type plotting capabilities should be identified as not having colour plotting capabilities.

11.3 Switching Between Graphics Mode and Alphanumeric Mode

The PLOTVX library subroutine TERCTN was modified to solve the problem of switching between graphics mode and scrolling mode. The entry point of 9 in the TERCTN subroutine was restructured to remove the dependency on the PLOT10 subroutine ANMODE. The function of this subroutine was to switch to alphanumeric mode. This subroutine was not performing consistently on all terminals so a new series of escape sequences to perform this function was implemented. This development was performed in collaboration with David Heath of DREA. This new capability has been tested on the Tektronix terminals and PC terminals equipped with the EM4010 emulation package and it appears to perform reliably.

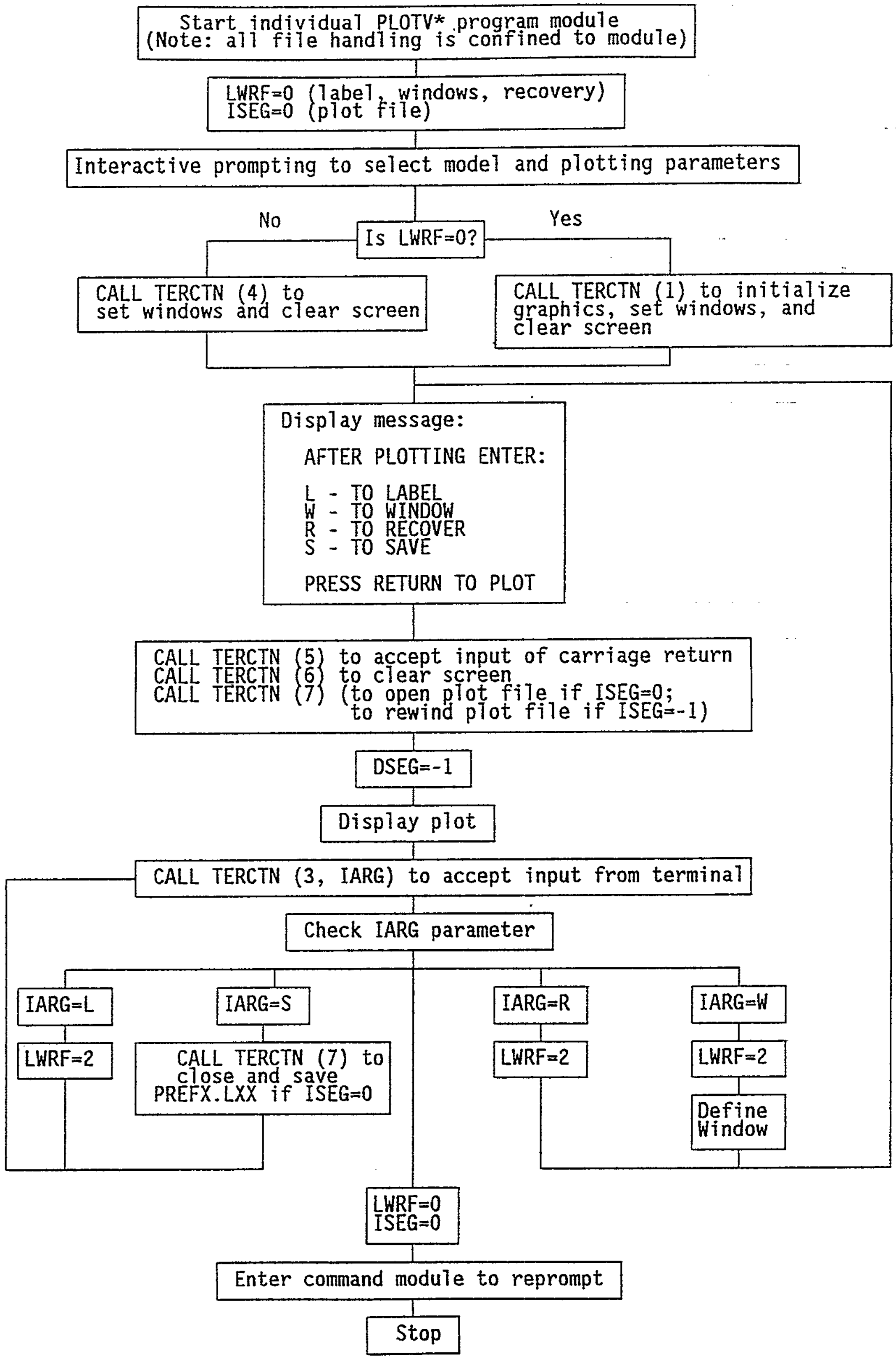


FIGURE 11.1: Procedure for Saving Plots for Laser Plotting

CHAPTER 12
MISCELLANEOUS VASTG ENHANCEMENTS

12.1 Introduction

The VASTG program is a graphics package for use with the VAST finite element program. It offers both pre-processing features, for verification of input data, and post-processing features, for interpretation and presentation of results. New features requiring major development effort are reported in separate chapters. Miscellaneous program enhancements developed under this contract are described below.

12.2 User Control on Elements Displayed in PLTV16

The PLTV16 program has been upgraded so that the user has control over the elements to be displayed. The PLTV16 program was modified to call the PLOTV library subroutine ELOPT which prompts for and fills an array with element plotting specifications. The PLTV16 program also required changes to the to use these plotting specifications.

12.3 Summary Table of Eigenvalues

The user has been provided with the capability to produce a summary of eigenvalues when prompting for mode to plot as required by contract item C-4. This feature was placed in PLOTV2 (plots deflected, mode shapes) and PLOTV9 (plots displacement or eigenmode contours). Two possible formats of the summary are shown below.

MODE NUMBER XX	FREQUENCY XXX (CPS)
.	.
.	.
.	.
.	.

or

MODE NUMBER XX	BUCKLING FACTOR XX
.	.
.	.
.	.

12.4 Improved Error Checking in Interactive Prompting of VASHID

Contract item C-5 was to improve error checks for interactive prompts of VASHID. This item was addressed by using the MARLIB library routines, READYN and READI. Subroutine READYN reads in Y or N response with error checking and subroutine READI reads in values of an integer I with error checking. The use of these standard routines ensures that the interactive prompting will not result in program crashes when incorrect responses are supplied. Incorrect responses now result in the user being reprompted.

12.5 Colour Reassignment Subroutine

Contract item C-15 required that the PLOTVX plotting library be expanded to include a colour reassignment subroutine. The reason for this requirement was to ensure the terminal is consistently initialized to the same colour mapping parameters prior to execution of VASTG. The colour mappings might not otherwise be set appropriately for the colour graphics options within VASTG. The colour mappings of a terminal may be changed by individuals or during the execution of various programs utilizing color graphics such as PATRAN.

The PL4113 plotting library has been studied to assess the capabilities of the existing routines. A subroutine called RESET which sets the screen mode to normal has been found. This subroutine will be implemented by a call in FACLTY if the user indicates that the terminal has colour capabilities. This subroutine should reset the terminal to the default colour menu (with index 0 transparent) as shown in Table 12.1. This table is provided to the user in the standard prompting of program prior to the input of colour index numbers. Additionally, the user has the capability to enter the command directive 'CMAP' to produce a Textronix colour map of the current terminal set-up. This colour map is a set of 16 colour blocks with their colour index numbers.

TABLE 12.1: Default Colour Code Index Numbers

1	WHITE	2	RED	3	GREEN
4	BLUE	5	CYAN	6	MAGENTA
7	YELLOW	8	ORANGE	9	GREEN-YELLOW
10	GREEN-CYAN	11	BLUE CYAN	12	BLUE-MAGENTA
13	RED-MAGENTA	14	DARK GREY	15	LIGHT GREY

CHAPTER 13

VAST DOCUMENTATION FILE

13.1 Introduction

A README.DOC file complete with installation instructions and a sample problem to be distributed with new releases of the VAST suite of program has been prepared. The contents of the README.DOC file are described in the following Section 13.2.

A set of twelve sample problems were developed under this contract to be used for validation of the VAST and VASTG codes. The description of these problems are described in the Section 13.3.

13.2 README.DOC File Contents

This file contains instructions for installing the VAS*.EXE programs on a Digital Equipment Corporation VAX/VMS system. Depending on the size of the tape dump, either BACKUP or COPY will be used for tape creation; BACKUP or COPY.

The contents of the tape will include: VAST60.EXE, VASTG6.EXE, VASGN8.EXE, and VASUSE.EXE. All the necessary object files will also be included so VAST06 and VASTG6 can be recreated with array dimensions increased or decreased as appropriate (see files VASTM.V60 and VASTG6.MTL containing FORTRAN source codes).

A sample TERMINAL_DEFAULTS.DAT file (see below) and some sample test problem files (including VAST input files [ASCII format] and solution files [binary format]). The graphics programs (all but VAST60.EXE) must be run on a Tektronix terminal (or Tektronix compatible terminal).

Interactive Execution of VASTG

The entire VAST program suite assumes that FOR005.DAT and FOR006.DAT are defined as SYS$INPUT and SYS$OUTPUT, respectively.

For interactive execution, the user should issue the following assignment statements:

```
$ ASSIGN SYS$INPUT FOR005
$ ASSIGN SYS$OUTPUT FOR006
```

Frequent users of VASTG may find it preferable to have these assignments made from within their LOGIN.COM file. These assignments have the effect of associating I/O from unit 5 to the default input stream, and I/O from unit 6 to the default output stream. In interactive mode, both streams default to the user's terminal.

Symbols

VMS provides local symbol definitions in order to simplify the loading of program images. For example in your LOGIN.COM file you could define simple execution commands as follows.

```
$ VAST60    :== RUN disk:[VAST60]VAST60.EXE
$ VASTG6    :== RUN disk:[VAST60]VASTG6.EXE
$ VASGN8    :== RUN disk:[VAST60]VASGN8.EXE
$ VASUSE    :== RUN disk:[VAST60]VASUSE.EXE
.
.
.
```

where "disk" is the name of the disk pack where the programs reside and [VAST60] is the name of the directory.

Terminal Defaults

A new feature of VASTG6 is the use of a file TERMINAL_DEFAULTS.DAT which should be located in your SYS$LOGIN: (i.e. home) directory. This file contains such information as your terminal type, preferred viewing specifications, whether you use a dialogue etc. This text file can be edited to allow you setup control with VASTG6. A sample file is as follows:

```
3 ! TEKTRONIX colour terminal (4100/4200), 2 for TEKTRONIX monochrome (4014...)
0 ! with no dialogue, 1 for dialogue
1 ! 0 for prompting, 1 for command driven version
0 ! 0-NIL, 1-IN., 2-FT.,  3-MM., 4-M.                     LENGTH UNITS
0 ! 0-NIL, 1-LBS, 2-KIPS., 3-N.,  4-KN., 5-MN.            FORCE UNITS
0 ! 0 for viewing vectors, 1 for angular rotations
```

Put a -1 for any of the last four entries if you want to be interactively prompted for a response.

<u>Example Problem</u>

The files GTES1.* are a set of input files into the Solver (VAST60.EXE) and resulting output files for post-processing by VASTG6.EXE. They can be used to quickly verify correct installation of the VAST codes.

<u>Recreating VAST Programs</u>

In the event that one has not requested the object files of the VAST programs, the following section is not relevant and therefore can be skipped.

This section contains the necessary instructions about recreating VASTG6 on a VMS based VAX. The following restrictions apply:

1) The VMS Version must be Release 5.0 or greater;

2) The VMS Fortran compiler must be Release 5.0 or greater; and

3) You must have the PLOT10 libraries available for linking (except for VAST60.-EXE recreation).

It is assumed that all "object" code modules necessary to relink VASTG6.EXE and the Fortran Source code of the "Driver" program "VASTG6.MTL" have been supplied. As indicated in the VASTG6 Reference Manual, the VASTG.MTL program can be edited to increase or decrease the limits on the size of the model to be displayed.

After the required modifications have be made to this file, rename the old executable image and recompile the "Driver" with the following commands.

```
$ RENAME VASTG6.EXE VASTG6_OLD.EXE
$ FORTRAN/NOOPTIMIZATION VASTG6.MTL
```

After the recompilation has taken place, relink the "object" modules with the following command.

```
$ @VASTG6.LNK
```

The new image file is called VASTG6.EXE.

Notes:

1) The command file "VASTG6.LNK" will have to be modified to correctly identify the location of relevant PLOT10 libraries (site specific).

2) If the array space of VASTG6.MTL is greatly increased, the PAGEFILE, and or BYTE limits of the account may have to be increased to handle the increased image size.

3) The image should not be made unnecessarily large as overall system performance may decrease needlessly through page faulting.

PC-Version

The contents of the disks will include: VAST60.EXE, VASTG6.EXE, VASGN8.EXE, VASTBC.EXE, VASLOD.EXE, and VASUSE.EXE.

A sample TERMINAL.DAT file (see below) and some sample test problem files (including VAST input files [ASCII format] and solution files [binary format]).

Terminal Defaults

A new feature of VASTG6 is the use of file TERMINAL.DAT which should be located in the root directory of C drive. This file contains such information as your terminal type, preferred viewing specifications, whether you use a dialogue, etc. This text file can be edited to allow you setup control with VASTG6. A sample file is as follows:

1 | 0 for prompting, 1 for command driver version
0 | 0-NIL, 1-IN., 2-FT., 3-MN., 4-M. LENGTH UNITS
0 | 0-NIL, 1-LBS., 2-KIPS., 3-N., 4-KN., 5-MN. FORCE UNITS
0 | 0 for viewing vectors, 1 for angular rotations

Put a -1 for any of the last four entries if you want to be interactively prompted for a response.

Example Problem

The files GTES1.* are a set of input files into the Solver (VAST60.EXE) and resulting output files for post-processing by VASTG6.EXE. They can be used to quickly verify correct installation of the VAST codes.

CHAPTER 14
SAMPLE PROBLEM SET

14.1 Introduction

Twelve different sample problems have been generated for the purpose of illustrating and verifying the plotting capabilities of recently enhanced VASTG program. These example problems have involved both unsubstructured and substructured finite element models. Analysis types include static, dynamic, natural frequency, buckling (stability), response spectrum, and frequency response. Table 14.1 provides a summary of the test problems. The data has been prepared appropriately for Version 6 of VAST.

14.2 Model Construction

A thin plate of rectangular shape is considered. The dimensions of the plate are 90 in. by 90 in. its thickness is 0.25 in. The plate is made of steel and the material properties were therefore defined as follows: Young's modulus = 30×10^6, Poisson's ratio = 0.300, and Density = 0.733×10^{-3}.

14.2.1 Unsubstructured Models

The plate is meshed with 3 x 6 thick/thin shell elements (IEC=1) with modelling option 1. Therefore, the .GOM file for these different types of analyses would be identical except for the .GOM file for buckling analysis, where the known initial stresses are applied on the structure. The parameter NSS in the GOM file is set equal to 0. The finite element mesh is shown in Figure 14.1.

14.2.2 Substructured Models

Two substructures are modelled using 3 x 3 mesh of thick/thin shells (IEC=1, modelling option 1). For simplicity, these two substructures are modelled identically in all aspects. Their dimensions are 90 in. in width and 45 in. in breadth and the thickness is 0.25 in. Parameter NSS is set equal to 2 in the GOM file. One superelement is defined from each of the two substructures. Master nodes are associated with the common interface and around the boundary of the plate. The SED file identifies the master nodes. The finite element mesh is shown in Figure 14.2.

14.2.3 Boundary Conditions

To model the simply supported edges, the nodes on the sides are constrained in translation in the z-direction. The nodes at the four corners of the rectangular plate are constrained in translations in y-, and z-directions as well. For substructured models, the boundary conditions are defined on master nodes. The boundary conditions are shown in Figures 14.1 and 14.2 for unsubstructured and substructured models, respectively.

14.3 Static Analysis - Unsubstructured Model

The geometry, material properties and boundary conditions for the square unstiffened plate were defined previously. Static pressure load of magnitude 1 lb/sq.in. was applied on two of the shell elements (see Figure 14.3). The prefix associated with this problem is GTAO1.

14.4 Dynamic Analysis - Unsubstructured Model

The geometry, material properties and boundary conditions for the unstiffened plate were defined previously. As time varying concentrated load (see Figure 14.4) is applied at node 37. The prefix for this problem is GTAO2.

14.5 Natural Frequency Analysis - Unsubstructured Model

The geometry, material properties and boundary conditions for the unstiffened plate were defined previously. No load is required for this type of analysis. The prefix for this problem is GTAO3.

14.6 Buckling Analysis - Unsubstructure Model

The geometry, material properties and boundary conditions for the unstiffened plate were defined previously. Preloading of 0.1 lb/sq.in. is applied on the structure in the x-direction to calculate the critical buckling load. The prefix for this problem is GTAO4.

14.7 Response Spectrum Analysis - Unsubstructured Model

A flat unstiffened steel plate is simply supported on all sides, but hinged at the four corners. The plate is made of steel with material properties as defined previously. The dimension of the steel plate is 90 in. in width and in breadth and the thickness is 0.25 in.

Loads for Response Spectrum Analysis are applied as a displacement. Since we are interested in investigating the response of the structure with respect to a spectra loading that coincides with the first seven harmonics of that structure. VASTV60 must then be run in two stages: (1) to determine the first seven mode shapes of the structure; and (2) to determine the response of the structure to a load spectra of these frequencies. The prefix for this problem is GTAO5.

14.8 Static Analysis - Substructured Model

A flat unstiffened plate is simply supported on all sides, but hinged at the four corners. The material properties of this rectangular plate are defined previously. Substructuring is used in this analysis. As described earlier, this involves two superelements being created from two substructures. Master nodes are defined on the interface between superelements and also at all nodes where boundary conditions are applied.

Static pressure load of magnitude 1 lb/sq.in. is applied on element 8 of superelement # 1 and element 2 of superelement # 2 (see Figure 14.5). The prefix for this problem is GTBO1.

14.9 Dynamic Analysis - Substructured Model

A flat unstiffened plate is simply supported on all sides, but hinged at the four corners. The material properties of this rectangular plate are previously defined. Substructuring is used in this dynamic analysis using direct integration method.

In this analysis, dynamic concentrated load is applied at the regular node 37 of substructure 1 (see Figure 14.6). The prefix for this problem is GTBO2.

14.10 Natural Frequency Analysis - Substructured Model

A flat unstiffened plate is simply supported on all sides, but hinged at the four corners. The material properties of this rectangular plate are defined earlier. Substructuring is used in this analysis. No load is require of for this type of analysis. The prefix for this problem is GTBO3.

14.11 Buckling Analysis - Substructured Model

A flat unstiffened plate is simply supported on all sides, but hinged at the four corners. The material properties of this rectangular plate are defined earlier. Substructuring is used in this analysis.

Preloading of 0.1 lb/sq.in. is applied on the structure in the x-direction to calculate the critical buckling load. The prefix for this problem is GTBO4.

14.12 Response Spectrum Analysis - Substructured Model

A flat unstiffened plate is simply supported on all sides, but hinged at the four corners. The material properties of this rectangular plate are defined earlier. Substructuring is used in this analysis. The prefix assoicated with this problem is GTBO5.

TABLE 14.1: Sample Problems for Validation of VASTG Graphics Programs

PREFIX OF SAMPLE PROBLEM	MODEL TYPE	ANALYSIS TYPE
GTA01	Unsubstructured	Static
GTA02	Unsubstructured	Dynamic
GTA03	Unsubstructured	Natural Frequency
GTA04	Unsubstructured	Buckling
GTA05	Unsubstructured	Response Spectrum
GTA06	Unsubstructured	Frequency Response
GTB01	Substructured	Static
GTB02	Substructured	Dynamic
GTB03	Substructured	Natural Frequency
GTB04	Substructured	Buckling
GTB05	Substructured	Response Spectrum
GTB06	Substructured	Frequency Response

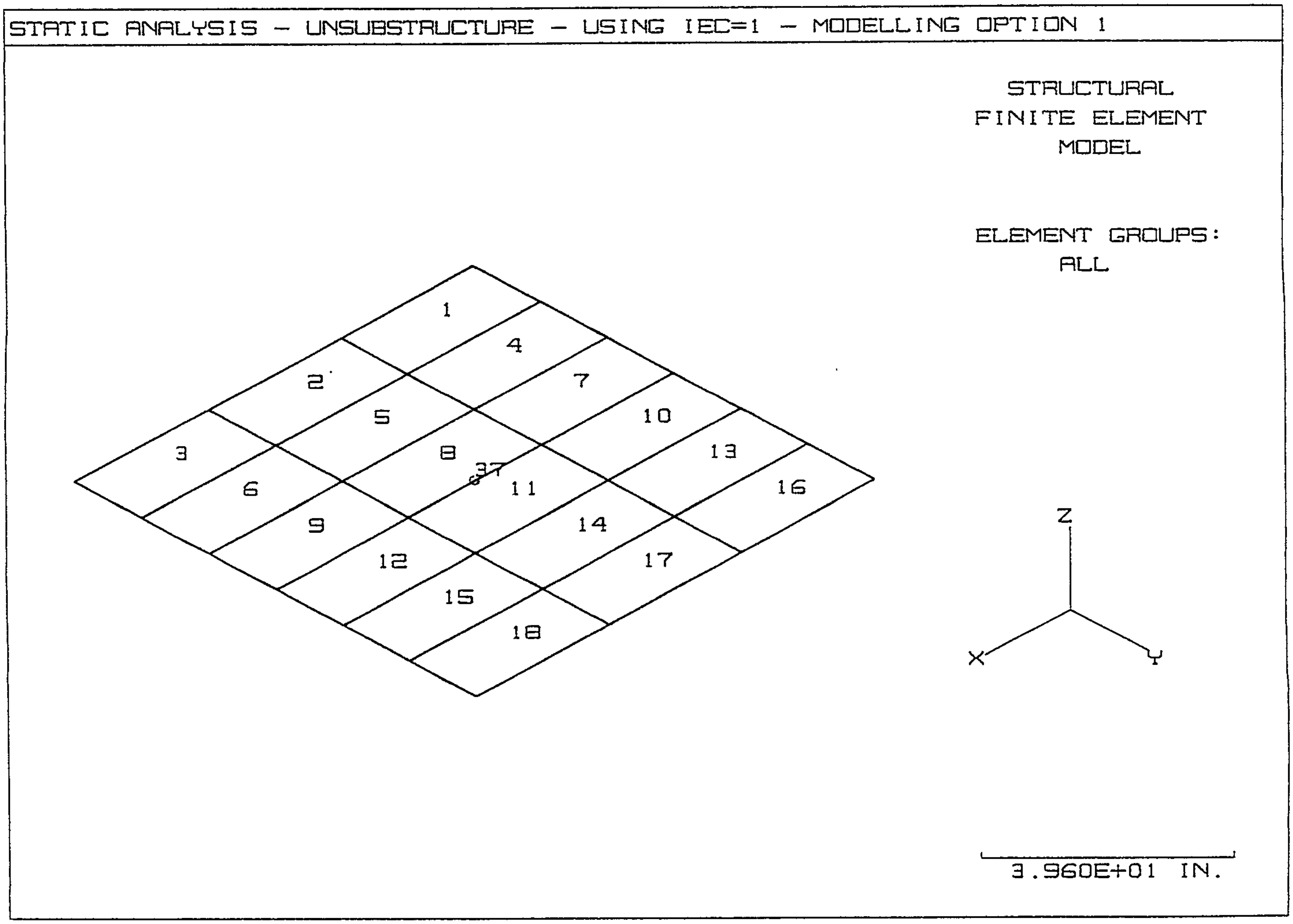


FIGURE 14.1: Unsubstructured Finite Element Test Model

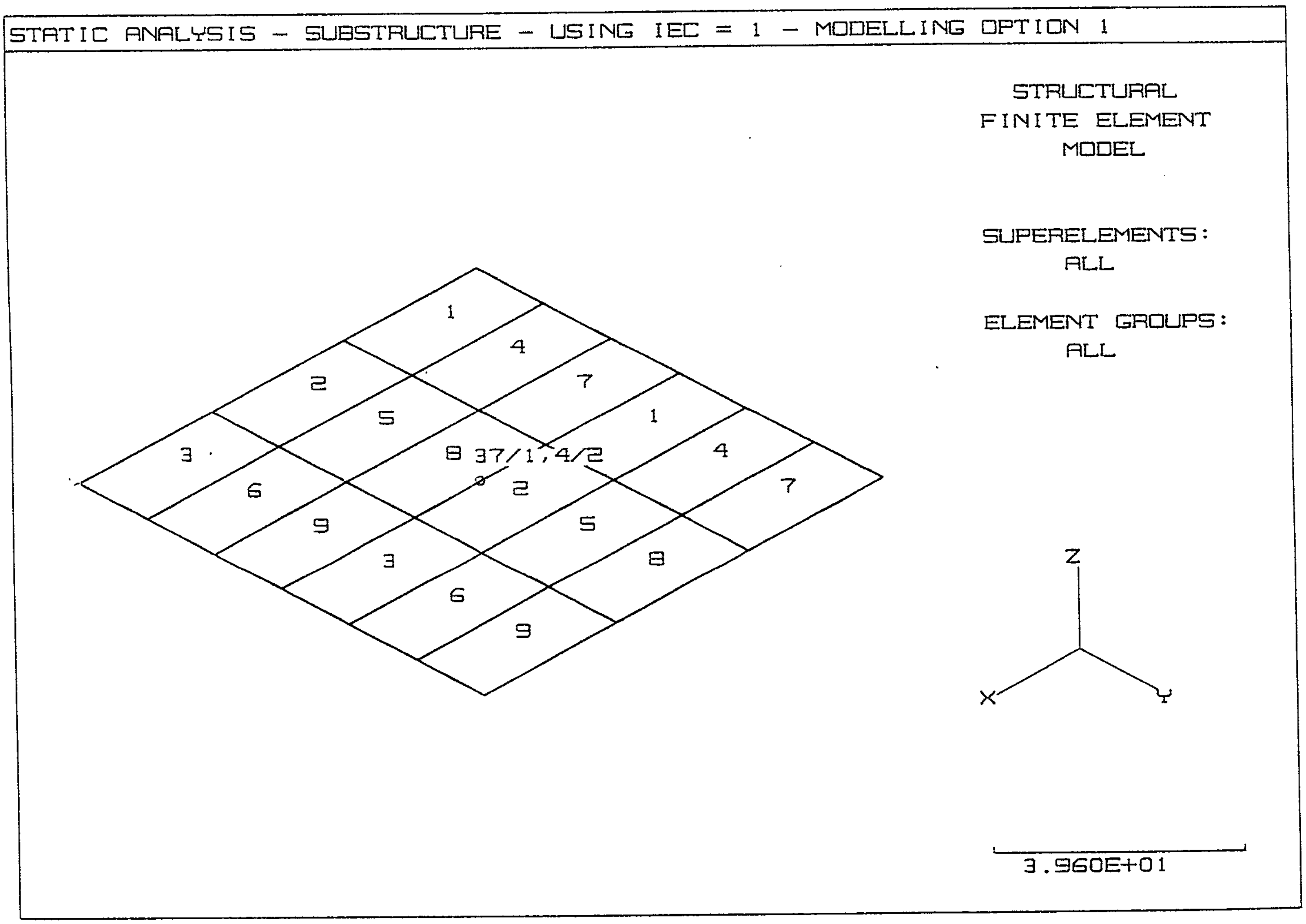


FIGURE 14.2: Substructured Finite Element Test Model

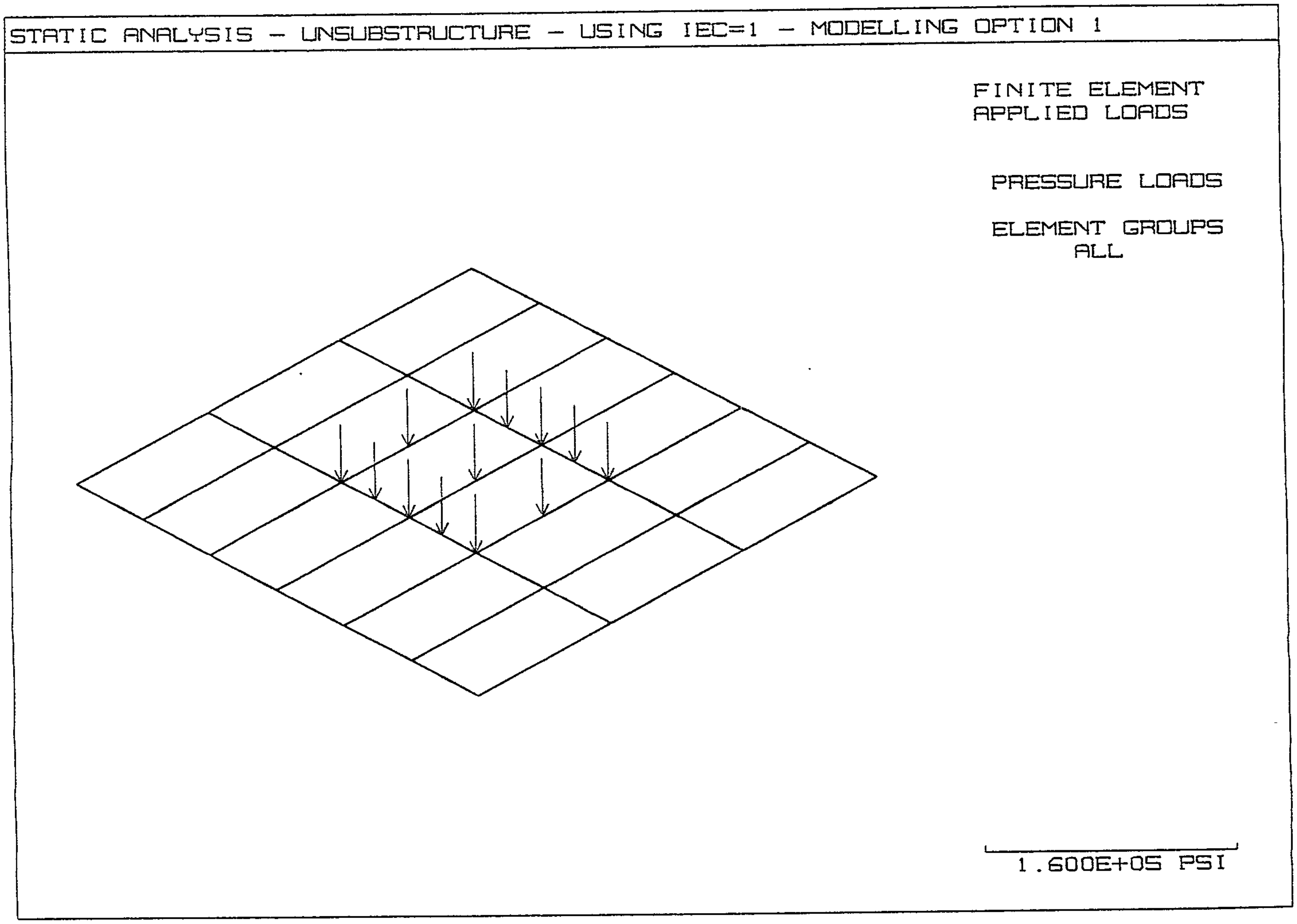


FIGURE 14.3: Static Pressure Load for the Unsubstructured Test Model GTA01

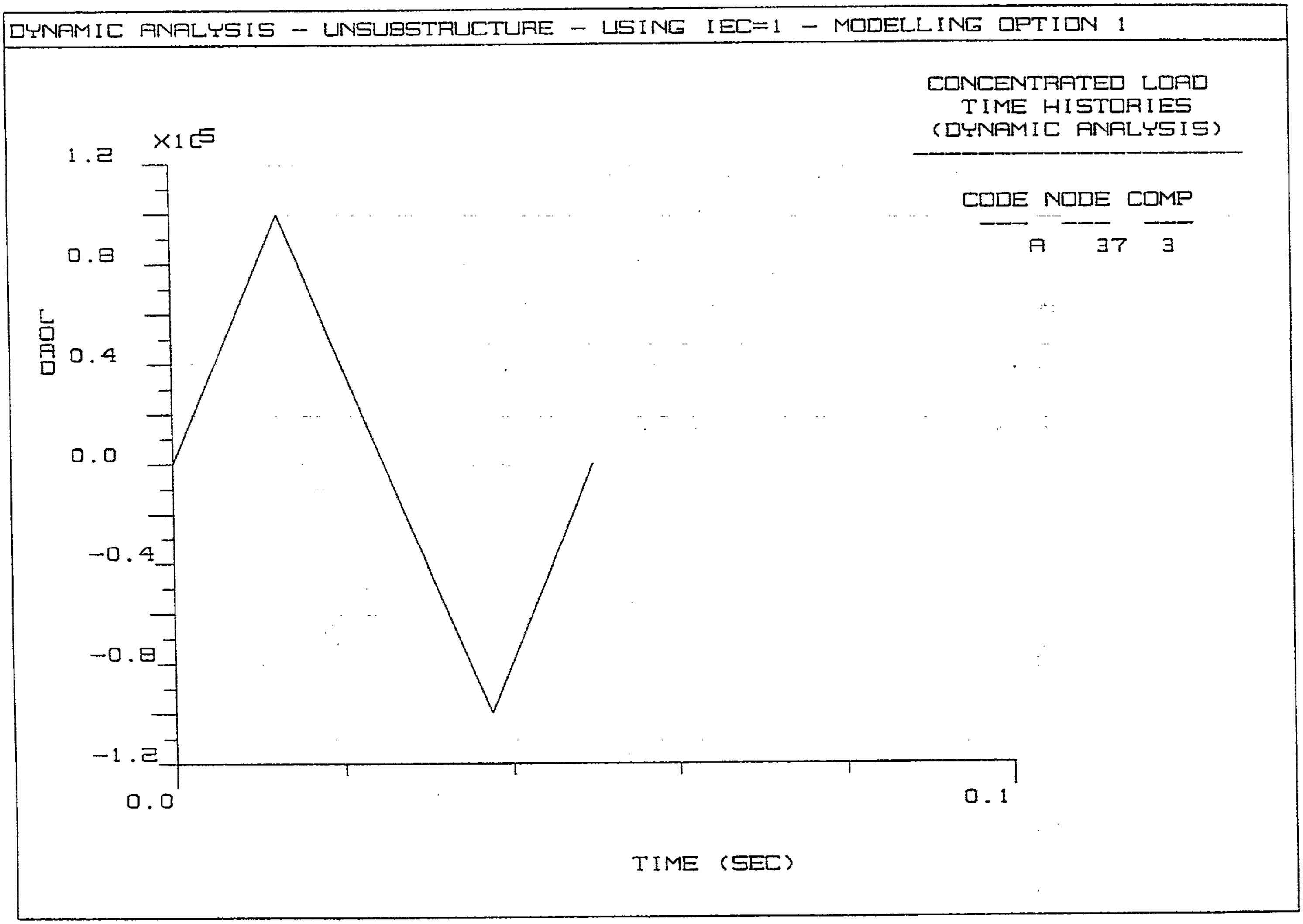


FIGURE 14.4: Dynamic Point Load for Node 37 for the Unsubstructured Test Model GTA02

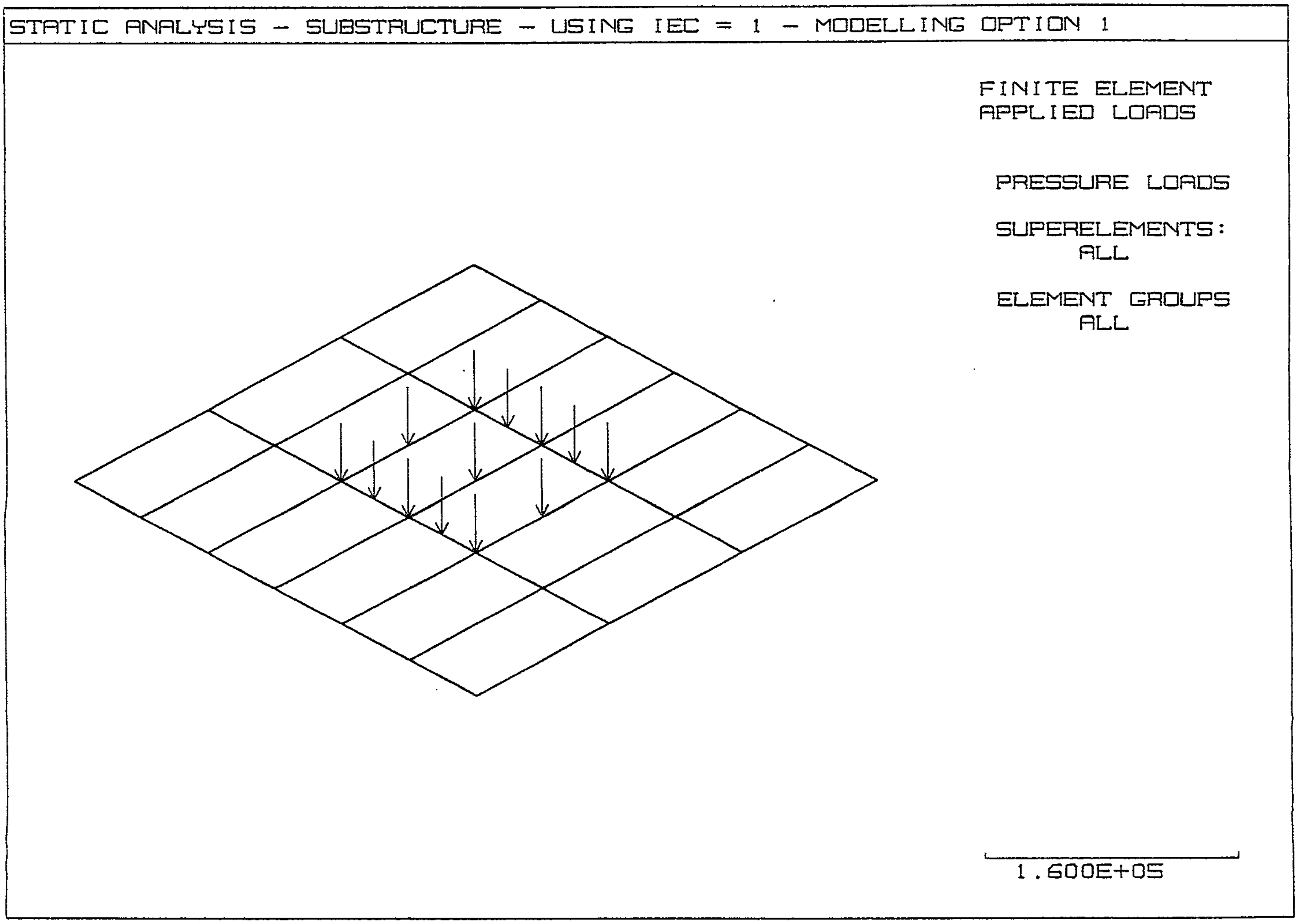


FIGURE 14.5: Static Pressure Load for the Substructured Test Model GTB01

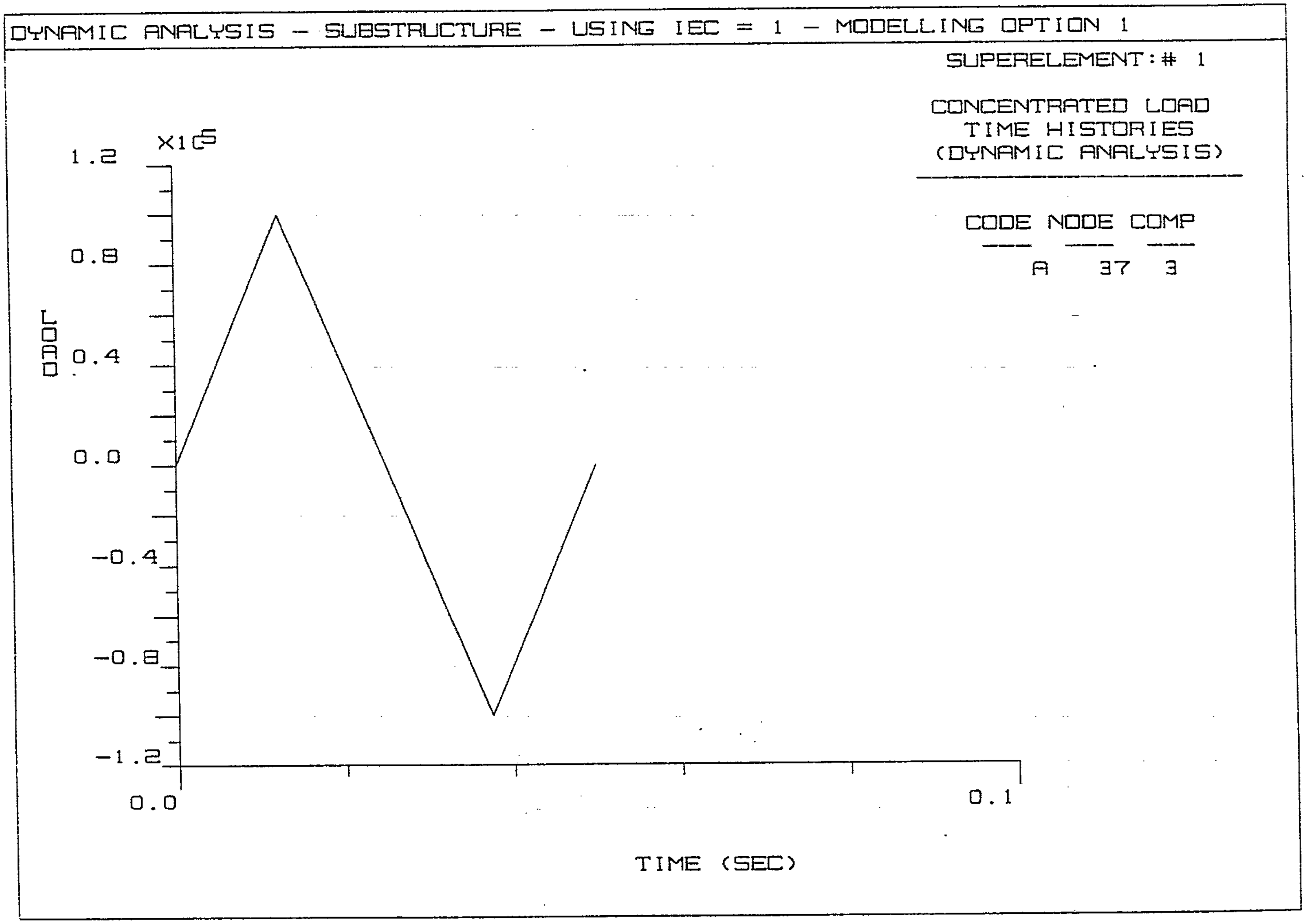


FIGURE 14.6: Dynamic Point Load for Node 37 of Substructure #1
for Substructured Test Model GTB02

CHAPTER 15
VASTG MANUAL

15.1 Introduction

The User's manual for the computer graphics program VASTG has undergone considerable changes under this contract both with respect to content and appearance. The contents of the manual has been updated to describe new features recently added to VASTG and are summarized in the following section. Missing operating instructions for VASTG program modules were added. The physical appearance of the manual has been upgraded through the adoption of an improved word processing package as discussed in Section 15.4. This can be readily confirmed by perusing the new manual.

15.2 New Features in Current Version of VASTG

The manual has been upgraded to include operating instructions associated with new features of VASTG:

1. The VASTG graphics program modules have been redesigned to be able to optionally run using single word command directives for the model plotting programs. The time history plotting programs do not yet have this feature. See Table 15.1 for a typical list of available command directives.

2. The VAST graphics program modules now have the capability to use default specifications. Selection of this option allows the individual program modules to operate using a session file containing the default specifications rather than by interactive prompting. See Table 15.2 for a sample of a typical session file responses.

3. The capability to produce colour plots has been implemented. The user may identify by colour the element types, element groups or superelements contained in the model.

4. The VAST graphics modules PLOTV3 and PLOTV4 which plot contours for stress/strain data and for displacement data, respectively, now automatically define element plotting specifications for visible surfaces.

5. The prompting related to units and to the view option have been placed in the main program. Individual plotting modules no longer prompt for this information.

6. The capability to scale the model plot to fill the screen when only part of the model is selected for viewing has been provided in the VAST graphics modules: PLOTV1, PLOTV2, PLOTV3, PLOTV4, PLOTV9 and PLOTV11.

7. The capability to use Z plane clipping has been provided in VAST graphics modules: PLOTV1, PLOTV2, PLOTV3, PLOTV4, PLOTV9 and PLOTV11. The user can specify two Z cutting planes which define a Z window for plotting. The Z coordinates corresponds to the coordinate normal to the screen and is dependent upon the view.

8. The stress (strain) post-processing program POSTV2 has the following features:

   i) processes maximum shear stress and strain data (referred to as SSM and ESSM on output); and
   ii) processes harmonic components of stress or strain for element models containing axisymmetric elements.

9. The following capabilities have been added to the pre-processing model display module, PLOTV1:

   i) plotting of shell normals and beam orientation vectors;
   ii) plotting of general beam cross-section data for user selected elements;
   iii) colour coding of elements by thickness; and
   iv) option to plot the isoparametric beam element as a solid.

10. The minimum and maximum values of stress(strain) are computed in PLOTV3 based upon the element plotting specifications.

11. The beam post-processing module, PLTV16, has a new feature to produce shear force and bending moment diagrams.

15.3 New Operating Instructions

The operating instructions section of the manual, for the PLOTHD module of the old version of VASTG, were revised to better reflect the program structure. The prompting within the VASTG program was also changed to refer to MOVIE rather than PLOTHD.

Operating instructions for the VASHID, VASHX, and VASHP programs were produced. The previous version of the manual did not contain any information on these programs.

15.4 Manual Appearance

The word processing package used to generate the VASTG manual has been changed. The manual now has a more professional appearance. The change over in the word processing package necessitated the retyping of tables. The opportunity was taken to review the table contents for consistency.

TABLE 15.1
COMMAND DIRECTIVES FOR PLOTV1

COMMAND DIRECTIVE	FUNCTION
HELP	To display this menu
EXIT	To terminate plotting
PLOT	To plot model
ELEM	To provide element plotting specs
COLR	To provide colour plotting specs
CMAP	To display colour map
VIEW	To provide view specs
NODE	To provide node plotting specs
ELNU	To have elements numbered
SHRI	To provide shrink specs
ZCLI	To provide z-plane clipping
BCON	To plot boundary conditions
LMAS	To plot lumped masses
SHEL	To plot shell normals
BEAM	To plot beam orientation vectors
SCAL	To scale plot to fill screen

TABLE 15.2

DEFAULT - SESSION FILE RESPONSES
FOR FINITE ELEMENT MODEL PLOTTING WITH PLOTV1

PARAMETERS	DEFAULT RESPONSE	DESCRIPTION
ILUNIT	0	Length units (default: nil)
IFUNIT	0	Force units (default: nil)
IVIEW	0	Viewing option (default: viewing vector approach)
IPNOD	0	Node plotting option (default: no nodes plotted
IANS	0	Element summary (default: no summary of elements)
IPELEM	1	Element plotting (default: all elements plotted)
NCOLR	0	Colour specifications (default: none provided)
DC1,DC2,DC3 RA,IGRF	1,1,1,0,10	Direction cosines for viewing vector option (default: isometric view)
ZR,YR,XR,IGRF	0,0,0,10	Rotations about body fixed axes for finite angular rotation option (default: view along z axes)
ISHRNK	0	Element shrink factor (default: no shrink)
IELNUM	0	Element numbering option (default: none)
INBN	0	Boundary condition display option (default: none)
NNBN	0	Boundary condition numbering (default: none)
INMS	0	Lumped mass display (default: none)

TABLE 15.2 (Continued)

DEFAULT - SESSION FILE RESPONSES
FOR FINITE ELEMENT MODEL PLOTTING BY PLOTV1

PARAMETERS	DEFAULT RESPONSE	DESCRIPTION
NNMS	0	Lumped mass numbering (default: none)
ISHELL	0	Shell normal display (default: none)
IBEAM	0	Beam orientation display (default: none displayed)
ISOLID	0	Curved beam plotting (default: plotted as line element)
IZCHK	0	Z-plane clipping (default: none)
IFILL	0	Plot scaling (default: not scaled to fill the screen)

REFERENCES

1. Vibration and Strength Analysis Program (VAST): User's Manual, Version #6.0. Martec Limited, Halifax, Nova Scotia (March, 1992).

2. Vibration and Strength Analysis Program (VAST-NL): User's Manual, Version #6.0. Martec Limited, Halifax, Nova Scotia (July 1992).

3. T.L. JANSSEN. A simple efficient hidden line algorithm, Computers and Structures, Vol. 17, No. 4, pp 563-571 (1983).

4. M.W. CHERNUKA et al.. "Nonlinear Static And Dynamic Buckling of Ship and Submarine Structures", Martec Technical Report TR-92-17 (SSC File No: OSC88-00189-013) (In preparation).

5. G.S. WATKINS. A Real-Time Visible Surface Algorithm. Computer Science Department, University of Utah, UTECH-CSc-70-101, June 1970.

6. H. CHRISTIANSEN and M. STEPHENSON. MOVIE.BYU Training Text. Community Press, Provo, Utah (1985).

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.) Martec Limited	2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable). Unclassified

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Enhancements of the VASTG Post-Processing and Graphical Display Programs

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Palmeter, M.F., Ooi, W.T., Chernuka, M.W.

5. DATE OF PUBLICATION (month and year of publication of document) September 1992	6a. NO OF PAGES (total containing information Include Annexes, Appendices, etc). 85	6b. NO. OF REFS (total cited in document) 6

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

DREA Contractor Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address).
Defence Research Establishment Atlantic
P.O. Box 1012
Dartmouth, NS, B2Y 3Z7

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant).	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written). W7707-0-1174
10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document).	10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor).

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected).

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

VASTG is an interactive graphics program designed specifically for use with the VAST finite element analysis program and it offers both pre-processing features for verification of the input data and post-processing features for interpretation and presentation of results. Numerous improvements have been made. The model verification modules PLOTVI and VASHID were upgraded to permit isoparametric curved beams to be plotted as solids and with eccentricities shown. VASHID was also improved in other respects to gain speed and reduce storage requirements. The load data verification capabilities of PLTV12 were expanded. The deformed model plotting capability utilizing VASHID was generalized to permit rotational degrees of freedom for the thick-thin shell elements to be either global or local. In addition, the automatic visible surface identification capability was extended to displacement and mode shape contours. Scaling on finite element plots was permitted to be automatically defined from element plotting specifications. Provisions were made to save graphics generated for terminal display on file for optional output to hardcopy devices such as laser printers. Furthermore, the capability for switching between graphics mode and alphanumerics mode was improved as well.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

finite element
ship structures
graphics

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM